sistema de archivos y administración de disco en linux

**ALUMNO: GONZÁLEZ JIMARÉZ ALEXIS SAÚL**

**SISTEMAS OPERATIVOS GRUPO: 06**

**PROFESORA: LAURA SANDOVAL MONTAÑO**

**ÍNDICE**

**SISTEMA DE ARCHIVOS……………………………………………………….2**

**ESTRUCTURA LÓGICA………………………………………………..2**

**NODO-I…………………………………………………………..2**

**GESTIÓN DE ALMACENAMIENTO SECUNDARIO………………..3**

**PERMISOS DEL SISTEMA SOBRE ARCHIVOS…………………….4**

**VARIOS SISTEMAS DE ARCHIVOS………………………………….4**

**PLANIFICACIÓN DE DISCO……………………………………………………7**

**GESTIÓN DE ALMACENAMIENTO DE DISCO……………………..7**

**CONCLUSIÓN……………………………………………………………………8**

**BIBLIOGRAFÍA………………………………………………………………….8**

**SISTEMAS DE ARCHIVOS.**

El sistema de archivos es la parte del núcleo encargada de gestionar los archivos del sistema. Entre sus funciones se tiene la creación de archivos, borrar archivos y directorios, la protección de la información, leer un archivo, escribir en un archivo, etc.

Las tareas o funciones que tiene el sistema de archivos es:

* Definir métodos de acceso
* Administración de archivos
* Administración de almacenamiento secundario
* Mecanismo de integridad de los archivos

**ESTRUCTURA LÓGICA.**

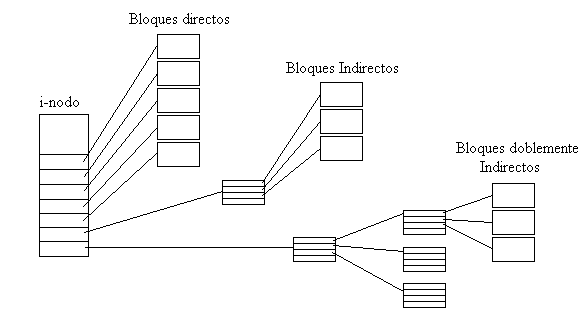
La estructura lógica que podemos encontrar dentro de un sistema de Linux es una estructura jerárquica, la cual es una estructura en árbol invertido en la que los archivos se agrupan en directorios. En él, todos los archivos y directorios dependen de un solo directorio denominado directorio raíz o *root*.

* NODOS-I

El nodo-i es la estructura de datos básica empleada por el sistema de archivos ext2 para localizar la información relativa a cada archivo o directorio.

Todos los archivos de Linux son administrados por el sistema operativo por medio de nodos-i. Un nodo-i (nodo índice) es una estructura de control que contiene la información clave de un archivo necesaria para el sistema operativo. Pueden asociarse varios nombres de archivo a un mismo nodo-i, pero un nodo-i activo se puede asociar con un único archivo y cada archivo es controlado por un solo nodo-i.

Los atributos de archivo, así como sus permisos y otra información de control, se almacenan en el nodo-i.



La información en un nodo-i residente en un disco de LINUX es:

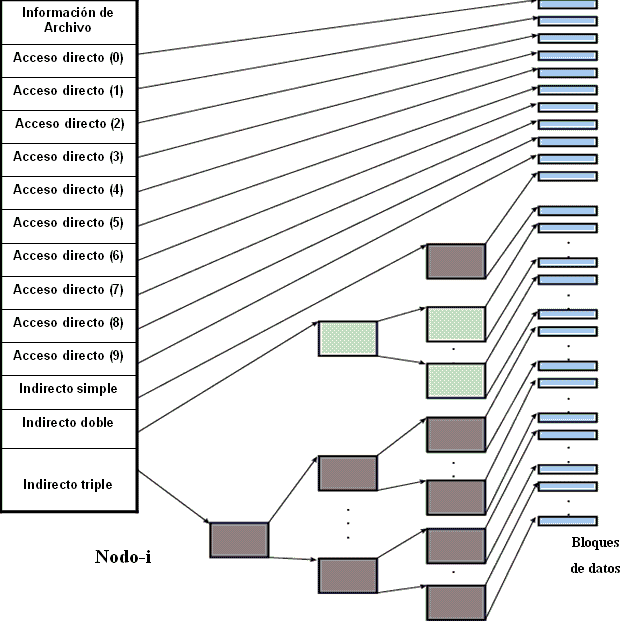
|  |  |
| --- | --- |
| Modo de archivo | Indicador (flag) de 16 bits que guarda los permisos de acceso y ejecución asociados con el archivo.  12-14 Tipo de archivo  9-11 Indicadores de ejecución  8 Permiso de lectura para el propietario  7 Permiso de escritura para el propietario  6 Permiso de ejecución para el propietario  5 Permiso de lectura para el grupo  4 Permiso de escritura para el grupo  3 Permiso de ejecución para el grupo  2 Permiso de lectura para el resto  1 Permiso de escritura para el resto  0 Permiso de ejecución para el resto |
| Cuenta de enlaces | Número de referencias al nodo-i en los directorios |
| ID del propietario | Propietario individual del archivo |
| ID del grupo | Grupo propietario asociado al archivo |
| Tamaño del archivo | Número de bytes del archivo |
| Direcciones del archivo | 39 bytes de información de dirección |
| Último acceso | Fecha del último acceso al archivo |
| Última modificación | Fecha de la última modificación del archivo |
| Modificación del nodo-i | Fecha de la última modificación del nodo-i |

**GESTIÓN DE ALMACENAMIENTO SECUNDARIO.**

En la memoria secundaria, un archivo consta de un conjunto de bloques. El sistema operativo o el sistema de gestión de archivos es responsable de la asignación de los bloques a archivos. Esto suscita dos cuestiones sobre la gestión en memoria secundaria. En primer lugar, debe asignarse el espacio de memoria secundaria a los archivos y, en segundo lugar, es necesario constancia del espacio disponible para asignar; es por ello que existen distintos métodos para la asignación en memoria secundaria.

Para el caso de Linux los archivos se asignan en bloques. La asignación es dinámica, a medida que se necesita. No se emplea asignación previa. Por lo tanto, los bloques de un archivo no tienen por qué estar contiguos necesariamente. Es por ello que dentro de este sistema operativo se utiliza un método de ***indexación*** para poder seguir la pista de cada archivo, con parte del índice almacenada en el nodo-i del archivo. El nodo-i incluye 39 bytes de información de dirección, organizada como 13 direcciones o punteros de 3 bytes. Las primeras 10 direcciones apuntan a los primeros 10 bloques de datos del archivo. Si el archivo es mayor de 10 bloques, se usan uno o más niveles de indirectos, de la forma siguiente:

* La dirección undécima del nodo-i apunta a un bloque del disco que contienen la siguiente parte del índice. Este bloque se conoce como el bloque indirecto simple. Este bloque contiene los punteros a los siguientes bloques del archivo.
* Si el archivo contiene más bloques, la dirección duodécima del nodo-i apuntará a un bloque indirecto doble. Este bloque contiene una lista de direcciones de bloques indirectos simples adicionales. Cada uno de los bloques indirectos simples contiene a su vez punteros a los bloques del archivo.
* Si el archivo contiene aún más bloques, la dirección decimotercera del nodo-i apuntará a un bloque indirecto triple que consiste en un tercer nivel de indexación. Este bloque apunta a bloques indirectos dobles adicionales.



**PERMISOS DEL SISTEMA SOBRE ARCHIVOS**

Como sabemos, Linux es un sistema multiusuario; como en un mismos sistema conviven varios usuarios, es importante implementar medidas que garanticen la seguridad de los archivos disponibles o bien de cualquier tipo de información en el sistema. Por dicho motivo, Linux, así como también la mayoría de los sistemas multiusuario, presenta un sistema de archivos y directorios. Las tres operaciones que se pueden realizar con un archivo o con cualquier tipo de directorio son:

* Leer
* Ejecutar (si es un ejecutable o un directorio)
* Escribir (incluye el borrar el propio archivo)

Como sabemos, cada archivo tiene un propietario, que inicialmente es el usuario que ha creado dicho archivo. El propietario de un archivo es el que marca los permisos del mismo, fijando quien puede acceder a dicho archivo, y qué se puede hacer con él. Esto es, si sólo permite que lo lean (y por lo tanto que lo puedan copiar), que lo ejecuten (si es un ejecutable, o un directorio para poder acceder a él), o que escriban en él (o bien lo borren).

**VARIOS SISTEMAS DE ARCHIVOS.**

Al introducir el sistema de archivos *ext,* fue necesario introducir un cambio fundamental en la estructura del sistema. Los sistemas de archivos reales fueron separados de la interfaz de llamadas al sistema por una capa denominada *Virtual File System (VFS).* VFS permite que Linux soporte diferentes sistemas de archivos, con una única interfaz de acceso a los servicios del sistema.

Las llamadas al sistema permiten realizar diversas operaciones sobre los archivos, tales como apertura, lectura, escritura, borrado o posicionamiento. En el nivel más cercano al hardware tenemos los manejadores de dispositivo o drivers. El manejador del dispositivo se encarga de ocultar al sistema de archivos las peculiaridades hardware del dispositivo físico donde se almacena la información de cada archivo. De este modo el sistema de archivos no sabe nada de pistas, cabezas o sectores, simplemente percibe el disco (o el dispositivo de almacenamiento secundario que se emplee como elemento de almacenamiento masivo) como una secuencia numerada de bloques de tamaño fijo.

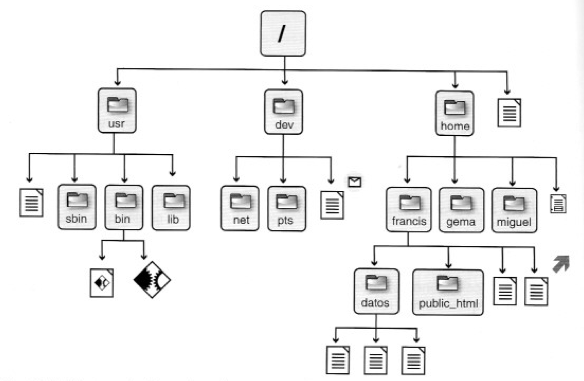
* Nodo-i VFS.

Cada archivo de cualquier sistema de archivos montado se representa por su correspondiente nodo-i VFS. La información almacenada en esta estructura de datos se genera a partir de rutinas específicas de cada sistema de archivos real. Estos nodos-i solo se mantienen en memoria y no tienen existencia física en ningún sistema de archivos real.

La información almacenada en dicha estructura, entre otras, es la siguiente:

* i\_dev: identificador del dispositivo donde se encuentra la información.
* i\_ino: número de nodo-i
* i\_mode: determina los derechos de acceso al archivo.
* i\_nlink: número de enlaces de archivo
* i\_uid: identifiacador de usuario propietario del archivo
* i\_gid: identificador del grupo al que pertenece el propietario
* i\_blksize: tamaño de bytes del bloque de datos
* i\_op: puntero a una estructura
* i\_count: número de componentes del sistema que están haciendo uso del nodo-i
* i\_lock: campo de bloque del nodo-i
* i\_dirt: campo que indica so el nodo-i ha sido modificado

Cada vez que se monta un nuevo sistema de archivos es necesario registrarlo y cada vez que se desmonte uno es necesario llevar a cabo la operación inversa. Los distintos sistemas de archivos registrados en el sistema aparecen en una lista enlazada que está apuntada por un puntero denominado file\_system.



**PLANIFICACIÓN DE DISCO.**

El planificador de disco en Linux, es el que se encarga de atender las peticiones de lectura o escritura, recolectadas y enviadas por el gestor de peticiones al disco en forma de bloques de caches, introducidas de forma ordenada en la cola de peticiones asignada a un dispositivo específico.

El algoritmo de planificación de disco utilizado por default en Linux es el C-Scan, pero si éste no cumple con las expectativas de una petición específica, una rutina verifica si existe otro algoritmo de planificación más adecuado para atenderla.

El Adaptive Disk Scheduling (planificador de disco adaptable), planifica las peticiones de e/s y las organiza en un horario priorizado de acuerdo a parámetros como retardo en la lectura, tiempo de búsqueda y preparación de disco, dependiendo de la prioridad y tipo de petición de e/s, el planificador de disco adaptable decide que petición debe ser atendida primero y que algoritmo de planificación se debe usar para atender la petición.

El objetivo que se persigue usando prioridades en las peticiones de e/s y la selección de algoritmos de planificación más adecuados, es que la cabeza lectora/escritora pueda moverse la menor cantidad de distancia para tener menor tiempo de respuesta a la atención de las peticiones de e/s.

**GESTIÓN DE ALMACENAMIENTO EN DISCO.**

Después de instalar Linux es posible:

* visualizar la tabla de particiones
* cambiar el tamaño de las particiones
* añadir particiones
* eliminar particiones

Dichas funciones se llevan a cabo mediante las utilidades “pated” o “fdisk”.

Existen diversas herramientas para el manejo de administración de disco, como son:

* Disk Druid: que es un programa editor de particiones y un administrador de sistemas de archivos.
* /Sbin/fdisk: el cual se utiliza desde la llínea de comandos y contiene picas opciones
* /Sbin/cfdisk: a diferencia de la anterior, en este caso presenta una utilidad gráfica.
* /Sbin/sfdisk: de igual manera se lleva a cabo mediante línea de comandos y contiene muchas opciones
* /Sbin/parted: utilidad que además de permitir la creación y eliminación de particiones, permite redimensionarlas.

Para el caso de “perted”, para poder usarla es necesario tener instalado el paquete parted, pero para poder crear, eliminar o cambiar el tamaño de una partición, el dispositivo en el que se encuentra no puede estar en uso.

Las tareas que se pueden realizar mediante parted, se ejecutan por línea de comandos:

|  |  |
| --- | --- |
| **COMANDO** | **DESCRIPCIÓN** |
| cp desde hasta | Copiar sistema de archivos entre particiones |
| help | Lista de comandos disponibles |
| mklabel | Crea etiqueta de disco para la tabla de particiones |
| mkfs numero-minor tipo-sa | Crea un sistema de archivos |
| rm numero-minor | Elimina la partición |
| select dispositivo | Selecciona un dispositivo diferente a configurar |
| print | Visualiza la tabla de particiones |

**CONCLUSIÓN:**

Como pudimos darnos cuenta en el sistema de archivos es de gran importancia determinar correctamente en base a lo que nosotros necesitamos el tipo de planificación que se va a llevar a cabo dentro del sistema, dicho método se determinara en base al tipo de sistema que se está manejando y del mismo modo sucede con la administración del o de los discos que se encuentran en el sistema, ya que como se mencionó, para el caso de administración en discos, este se va adaptando, dependiendo de la capacidad que se requiera, de tal forma que siempre se busca que los procesos que se llevan a cabo sean los más rápidos para que de este modo se pueda trabajar de una manera más eficiente.

**BIBLIOGRAFÍA:**

* [**http://es.tldp.org/Presentaciones/1999hispalinux/conf-jfernandez/7.pdf**](http://es.tldp.org/Presentaciones/1999hispalinux/conf-jfernandez/7.pdf)
* [**http://sopa.dis.ulpgc.es/ii-aso/portal\_aso/leclinux/administracion/discuota/disco.pdf**](http://sopa.dis.ulpgc.es/ii-aso/portal_aso/leclinux/administracion/discuota/disco.pdf)
* **Artículo visto en clase**