PRACTICA #5

"CONCEPTOS DEL SISTEMA DE ARCHIVOS"

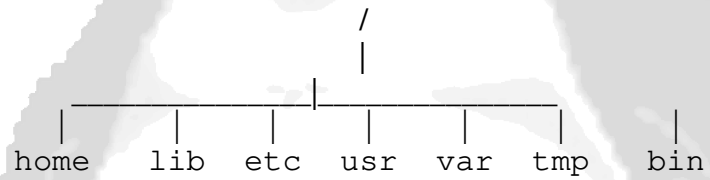
OBJETIVO: Confirmar los conocimientos del alumno acerca del sistema de archivos.

INTRODUCCION:

Un sistema de archivos es la estructura que permite organizar la información de LINUX.

El sistema de archivos cuenta con las siguientes características:

- Una estructura jerárquica arborescente, en donde el nodo principal es el directorio llamado raíz (representado por "/") y cada uno de los niveles del árbol representa directorios.



- Consistencia en el manejo de archivos.
- Protección para los datos de los archivos.
- Manejo de los dispositivos periféricos como archivos.

La estructura del sistema de archivos consiste en el bloque de boot, el super bloque la lista de i-nodos y el bloque de datos.

1.- El bloque de boot ocupa la parte del principio del sistema de archivo, por lo regular el primer sector, y puede contener el código de boot o de arranque.

2.- El superbloque describe el estado de un sistema de archivos. Contiene información acerca del tamaño de los sistemas de archivos, la lista de bloques libres disponibles, índice del siguiente bloque libre en la lista de los mismos, el tamaño, el total y la lista de los i-nodos, el índice del siguiente i-nodo libre, los campos de bloqueo de elementos de las listas de bloques libres y de i-nodos libres. Estos campos se emplean cuando se realiza una petición de bloque o de i-nodo libre, también contiene el flag que sirve para indicar si el superbloque ha sido modificado o no.

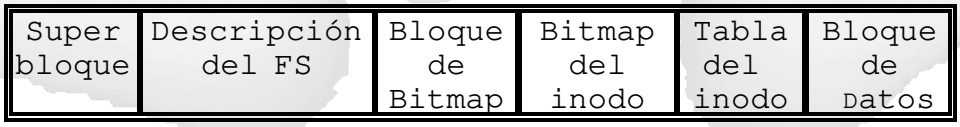
3.- La lista de i-nodos, la cual se encuentra a continuación del superbloque. En esta lista, durante el proceso de arranque del sistema, el kernel lee la lista de inodos del disco y carga una copia en memoria, conocida como tabla de i-nodos. Las manipulaciones que hace el subsitema de archivos involucran a dicha tabla, pero no a la lista de i-nodos. Mediante éste mecanismo se consigue una mayor velocidad de acceso a los archivos, ya que la tabla se encuentra cargada siempre en la memoria. Los campos de un i-nodo son los siguientes:

a. Identificador del propietario del archivo.
b. Tipo de archivo.
c. Tipo de acceso al archivo.
d. Tiempos de acceso al archivo.
e. Número de enlaces del archivo.
f. Entradas para los bloques de dirección.
g. Tamaño del archivo.

La tabla de i-nodos contiene la misma información que la lista de i-nodos, además de la siguiente información adicional:

i. El estado del i-nodo, el cual indica si esta bloqueado, si hay algún proceso esperando a que el i-nodo se desbloquee, si la copia del i-nodo que hay en memoria difiere de la que está en el disco, si la copia de los datos del archivo que hay en memoria difiere de los datos que hay en el disco.
ii. El número de i-nodo, como los i-nodos se almacenan en disco en un arreglo lineal, al cargarlo en memoria, el kernel le asigna un número en función de su posición en el arreglo.
iii. Apuntadores a otros i-nodos cargados en memoria, el kernel enlaza los i-nodos sobre una cola hash (cola de dispersión) y sobre una lista libre. Las claves de acceso a la cola hash nos la dan el número de dispositivo lógico del i-nodo y el número de i-nodo.
iv. Un contador que indica el número de copias del i-nodo que están activas.

4.- El bloque de datos empieza a continuación de la lista i-nodos y ocupa el resto del sistema de archivos. En esta zona es donde se encuentra situado el contenido de los archivos a los que se hace referencia la lista de i-nodos. Cada uno de los bloques destinados a datos sólo puede ser asignado a un archivo, tanto si lo ocupa totalmente como si no.



Se tiene diseñado e implementado 2 nuevos sistemas de archivos y estos están incluidos en el estandar del kernel de Linux. Estos 2 llamados “Extended File System [extfs]’ y “Second Extended File System [ext2fs]” arregla las limitaciones y nuevas características.

Presentaremos la implsntsción del Sistema de Archivo Virtual en Linux y se detallará el Second Extended File System del código del kernel y las utilerías de modo usuario.

Después de la integración del VFS en el kernel, un nuevo sistema de archivos llamado “Extended File System” fue implementado en Abril de 1992 y agregado a Linux 0.96c. Este nuevo sistema de archivos remueve a las 2 grandes limitaciones de Minix. Máximo tamaño fue de 2 Gigabyte y el máximo nombre del archivo fue de 255 caracteres. Fue un mejoramiento sobre el sistema de archivos de Minix pero algunos problemas están presentes. Estos no soportan los accesos separados, modificando el inodo, y las modificaciones de datos están en tiempos estampados. El sistema de archivos usan ligas de listas a guardar track de bloques libres e inodos y están producen malos performances : como los sistemas de archivos fueron usados, las listas comenzaron sin orden y los sistemas comenzaron a fragmentar. Como una respuestas a estos problemas, 2 nuevos sistemas de archivos fueron soltados en la versión alfa de enero de 1993 : El sistema de archivos Xia y el Second Extended File System. El sistema de archivos Xia fue basado en el sistema de archivos Minix y el código del kernel únicamente agrega un poco improvisado sobre los sistemas de archivos. Básicamente, provisto nombre largo en los archivos, soporta grandes particiones desde 3 tiempos estampados. Por el otro lado, ext2fs fue basado con el código de extfs con mucha reorganización y mejoramiento. Tiene que estar diseñado con evolución en mente y contiene espacio a futuros mejoramientos. Este será descrito con mas detalle en páginas posteriores.

Cuando los 2 nuevos sistemas de archivos fueron primeros implementados, provistos esencialmente el mismo aspecto. Estos sistema de archivos diseñados, Xiafs fue mas estable que ext2fs. Como los sistemas de archivos fueron usados más, los bugs fueron fijos en el ext2fs y las mejora tiene nuevos aspectos integrados. Ext2fs es ahora muy estable y tiene comenzando el factor estandard del sistema de archivos de Linux.

Esta tabla contiene un sumario de los aspectos provistos de los diferentes sistema de archivos

	MINIX FS	EXT FS	EXT2 FS	XIA FS
Max FS Size	64 MB	2 GB	4 TB	2 GB
Max file Size	64 MB	2 GB	2 GB	64 MB
Max file name	16/30 c	255 c	255 c	248 c
3 time support	No	No	Yes	Yes
Extensible	No	No	Yes	No
Var. Block size	No	No	Yes	No
Maintained	Yes	No	Yes	?

El Sistema de Archivos Virtual (VFS)

El kernel de Linux contiene un VFS que usado durante la llamada al sistema accionando en archivos. El VFS es una capa indirecta cual tiene un archivo orientado a la llamada del sistema y llama necesariamente funciones en el sistema de archivos físicos codificados en el I/O. Este mecanismo indirecto es frecuentemente usado en los sistemas operativos UNIX, fácilmente a la integración y el uso de varios tipos del sistema de archivos. Cuando un proceso saliente a un archivo orientado a la llamada del sistema, la llamada del kernel como función contiene el VFS. Esta función contiene una estructura independiente manipulando y redireccionando la llamada a una función contenida en el código del sistema de archivos físicos, cual es la responsable de la cabecera a la estructura que dependen de operaciones. El código de sistemas de archivos usa el buffer cache correspondiendo a las funciones de I/O a los dispositivos.

La definición del VFS pone las funciones como un buen sistema de archivos que está implementado. Esta interfase hace una puesta de operaciones asociadas a 3 grandes objetos : sistemas de archivos, inodos y archivos abiertos. El conocido VFS acerca de los tipos de sistema de archivos soportados en el kernel. Usa una tabla definida durante la configuración del kernel. Cada entrada de esta tabla describe un tipo de sistema de archivos : Contiene el nombre del tipo del sistema de archivos y pone en una llamada de función durante el montaje de operación. Cuando un sistema de archivos es montado, el apropiado montaje de la función es la llamada. Esta función es responsable de leer el superbloque desde el disco, inicialmente variables internas y retornar un montaje del sistemas de archivos describe el VFS. Después que el sistema de archivos es montado, las funciones del VFS puede usarse una descripción accesando a la rutinas físicas del sistemas de archivos.

Un sistema de archivos montado describe el contenido de algunas de datos : información que es común son los tipos de sistemas de archivos, punteando funciones provistas por los sistemas de archivos físicos en el código del kernel, y datos privados mantenidos por el código físico de los sistemas de archivos. La función puntera contiene la descripción del sistema de archivos alojados en el VFS a accesar la rutinas internas del sistemas de archivos.

The Second Extended File System

El Second Extended File System (ext2fs) ha sido diseñado e implementado a fijar algunos problemas presentados en el primer extfs.

En el Sistema Operativo, necesitamos que el ext2fs tuviera un excelente performance. Se necesita que fuera un muy robusto sistema de archivos en orden a reducir los riesgos de datos perdidos en uso intensivo. Lo último es que el ext2fs tiene que incluir provisiones de extensiones a alojar usuarios a beneficiar desde nuevos aspectos sin reformas al sistemas de archivos.

Reformas del “Estandar” ext2fs

Soporta el ext2fs los estandares de los tipos de archivos UNIX : archivos regulares, Directorios, Archivos Especiales de Dispositivos, Ligas, Sockets, etc. Ext2fs es capaz de manejar sistemas de archivos creados en grandes particiones. El código del kernel original restringe el máximo tamaño de los sistemas de archivos a 2 GB. Recientemente de trabajo en la capa de VFS se tiene un limite de 4 TB. Es posible usar discos grandes sin necesidad de crear muchas particiones.

Ext2fs proviene de la longitud de los nombres de archivos. Usa longitud variable en la entrada de directorios. El máximo tamaño del nombre del archivo es de 255 caracteres. Este limite puede ser extendido a 1012 si es necesario. Ext2fs reserva algunos bloques del superusuario (root). Normalmente es del 5% los bloques reservados. Esto aloja al administrador a recobrar fácilmente situaciones cuando los procesos llenan el sistema de archivos.

Reformas “Avanzadas” del ext2fs

En suma al aspecto estandar UNIX, ext2fs soporta algunas extensiones cual no es usualmente presentado en los sistemas de archivos UNIX. Atribuidos al archivo aloja al usuario a modificar al kernel cuando acciona una puesta de archivos. Uno puede poner atributos a un archivo o a un directorio. En el último de los casos, nuevos archivos creados en el directorio inherentes a estos atributos. BSD o System V Release 4 pueden seleccionar como tiempo de montaje. Una opción de montaje aloja al administrador a escoger la semántica de la creación de archivos. En un sistema de archivos montados con semánticas BSD, son archivos creados con el mismo grupo id como es el directorio padre. System V está en su semántica son con un bit mas complejo : Si un directorio tiene puesto un bit setgid en nuevos archivos inherentes al grupo id del directorio y subdirectorios inherentes al grupo id y el setgid ; en el otro caso, archivos y subdirectorios son creados con el grupo primario id de los llamados procesos con el grupo primario id de los llamados procesos.

Una opción de montaje aloja al administrador a solicitar este metadato (inodos, bloques de bitmap, bloques indirectos y directos) ser descritos sincronizados en el disco con ellos modificándolos. Esto puede ser provechoso a mantener un estricto metadato consistente, pero esto liderean una pobre ejecución. Actualmente, este aspecto no es normalmente usado, desde entonces una suma al ejecutar una asociación pérdida con actualizaciones usando sincronización de los metadatos, puede causar corrupción en el uso de los datos cual no es bandera por lo cual el filesystem es checado.

Ext2fs aloja al administrador a escoger el bloque lógico de tamaño cuando crea el sistema de archivos. El tamaño de bloques puede ser típicamente de 1024, 2048 y 4096 bytes. Usando grandes tamaños de bloques pueden correr arriba de I/O desde pocas I/O de respuestas, y así pocos discos de cabeceras, la cual necesitamos al acceder a un archivo. En la otra cabeza grandes bloques desolados más espacio en disco : En el promedio, el último bloque alojado a un archivo es únicamente

medio lleno, así como bloques obtenidos grandemente, más espacio espera en el último bloque de cada archivo. En suma, mas de la ventaja de los tamaños de bloques lagos son obtenidos por el sistema de archivos con técnicas de prealojamiento.

La implementación de ext2fs de rápidas ligas simbólicas. Una liga simbólica rápida no es usada en un bloque de datos en el sistema de archivos. El nombre en blanco no esta almacenado en un bloque de datos pero en el inodo. Esta política nos salva de algún espacio en disco (no a bloques de datos que necesitamos a ser alojados) y velocidades arriba de las operaciones ligadas (Esto no es necesario en leer en un bloque de datos cuando se accede mucho a una liga), el espacio disponible en el inodo es limitado y no esta todo ligado o puede ser implementado como una liga simbólica. El tamaño máximo del nombre en blanco es un símbolo de ligado rápido de 60 caracteres, planeamos a extender este esquema de archivos a pequeños archivos en un futuro cercano.

Ext2fs guarda el track en el estado de los sistemas de archivos. Un campo especial en el superbloque, es usado por el código del kernel que indica el estado del sistema de archivos. Cuando un sistema de archivos es montado en modo de lectura/escritura , este estado esta puesto ó indicado como “no limpio”. Cuando es desmontado o remontado en un inodo en un modo solamente_lectura, este estado es reseteado como “limpio”. Como en el tiempo de boot, el sistema de archivos checa usando esta información y decide si el sistema de archivos es checado. El código del kernel también graba errores en estos campos. Cuando una inconsistencia esta detectado por el código del kernel, el sistema de archivos es marcado como “erróneo”. El sistema de archivos checa con un tets y forza el chequeo del sistema de archivos indiferente de estos aparentes estados “limpios”.

Siempre que el sistema de archivos salta a checar en algún tiempo si esta dañado, así el ext2fs provee 2 caminos para hacer el chequeo forzado en intervalos regulares. Un contador de montaje es mantenido en el superbloque. Cada tiempo el filesystem es montado en modo lectura/escritura, este contador es incrementado. Cuando este es valuado en máximo (también grabado en el superbloque), el filesystem checa forzadamente y hace que el filesystem este “limpio”. Un último chequeo de tiempo y un máximo de chequeo de intervalo o son también mantenidos en el superbloque.

Estos 2 campos alojados al administrador a solicitud del chequeo periódicamente. Cuando el máximo chequeo del intervalo ha sido alargado, el checador ignora el estado del filesystem y forza el checado del mismo. Ext2fs ofrece utilerías al comportamiento del filesystem. El programa tunefs puede ser usada a modificar :

I. El error de comportamiento. Cuando una inconsistencia es detectada por el código dl kernel, el filesystem es marcado como “erróneo” y uno de las 3 siguientes acciones puede ser acabado : continua una normal ejecución, remonta el filesystem en modo solamente_lectura a evitar el filesystem corrupto, hacer pánico en el kernel y rebotear, correr el chequeo del filesystem.

I I .

El contador máximo de montaje.

III. El chequeo máximo del intervalo
IV. El no. de bloques lógicos reservados para el superusuario.

Opciones de montaje puede también ser usado a cambiar el comportamiento erróneo del kernel. Un atributo alojado al usuario solicita seguridad borrado en archivos. Cuando cada archivo es borrado, el dato aleatorio está escrito en el bloque del disco previo alojado en el archivo. Esta prevención maliciosa de gente desde la ganancia en el acceso previo al contenido de archivo usando un editor de disco.

Ultimo, nuevos tipos de archivos inspirados desde la versión 4.4 BSD del filesystem que está recientemente agregado a ext2fs. Archivos inmutables pueden únicamente ser leídos : nadie puede escribir o borrar esto. Esto puede ser usado para proteger configuraciones de archivos sensitivos. Archivos únicamente agregados puede ser abiertos en modo escritura pero los datos está siempre agregando al final del archivo. Archivos similarmente inmutables, estos no pueden ser borrados o renombrados. Esto es especialmente útil a los archivos, los cuales pueden únicamente crecer.

CUESTIONARIO:

1.- ¿Cuáles son las características del Sistema de Archivos?
2.- ¿Cuál es la estructura del Sistema de Archivos?
3.- ¿Qué información se encuentra en los i-nodos?
4.- ¿Qué información contiene el Super Bloque?
5.- ¿Cuáles son las caracteristicas del Virtual File System (VFS)?
6.- ¿Cuáles son las diferencias y semejanzas del extfs y ext2fs?
7.- ¿Que diferencia notable tiene ext2fs y minix?
8.- ¿Cuál es la capacidad máxima de un archivo con el formato ext2fs?

CONCLUSIONES: