



Brayan Josue Reyes Salazar

Unidad5

“lectura”

M.S.C Eduardo Flores Gallegos

05-06-18

El camino de unix

Linux analiza su sistema de archivos de la misma forma que Unix lo hace, adoptando los conceptos de superbloque, inode, directorio y archivo. El árbol de archivos accesibles en cualquier momento está determinado por la forma en que se ensamblan las diferentes partes, cada parte es una partición del disco duro u otro dispositivo de almacenamiento físico que está "montado" en el sistema

El modo unix

Este analiza la información adoptando los conceptos de súper block, inodos, directorios y archivos. En cada parte es una partición del disco duro u otro dispositivo de almacenamiento físico que se esté utilizando en el sistema.

Las definiciones de esos conceptos para comprender más a fondo el tema:

El Súper Block debe su nombre a su herencia, desde que se usó el primer bloque de datos de un disco o partición para contener meta información sobre la propia partición.

Un i-nodo está asociado a cada archivo, El propietario, el grupo, los permisos y los tiempos de acceso de un archivo se almacenan en su i-nodo, así como el tamaño de los datos que posee, el número de enlaces y otra información.

El directorio es un archivo que asocia los inodos a los nombres de archivo. Las funciones específicas de cada tipo de sistema de archivos se utilizan para leer y modificar el contenido de un directorio independientemente del diseño real de sus datos.

Los archivos son áreas de datos, pero también pueden ser directorios, dispositivos, FIFOs (primero en la primera salida) o Sockets.

Un sistema operativo debe ser capaz de hacer frente a diferentes maneras de diseñar información en disco. un objeto declara un conjunto de operaciones que se deben utilizar para tratar con ella. El kernel no se pegará en las sentencias de switch grandes para poder acceder a los diferentes esquemas físicos de los datos,

y se pueden agregar y eliminar nuevos tipos de sistemas de archivos en tiempo de ejecución.

El VFS implementa un conjunto de operaciones para actuar en los objetos, la mayoría de las operaciones estas a su vez reciben un puntero al objeto self, así permitiendo la modificación.

La implementación de Linux permite el uso de módulos para todo tipo de archivos, el sistema root debe de ser montado antes de cargar un módulo. Pero esta operación solo puede ser montada en disquetes.

Los números de i-nodo se almacenan en la estructura de datos asociada a cada archivo dinámico en donde el sistema de archivos no tiene ningún dispositivo físico para extraer sus datos y así asignar números de i-nodo asignados dinámicamente.

La mayoría de las funciones de VFS como todas las llamadas de sistema y otras funciones del núcleo devuelven un cero o un número positivo en caso de éxito y un número negativo en caso de errores, los códigos de error devueltos por las funciones del kernel son siempre uno de los valores enteros definidos en

Una de estas estructuras está asociada a cada nodo dentro de /proc, y se utiliza para mantener un seguimiento del árbol de archivos. Una de estas estructuras está asociada a cada nodo dentro de /proc, y se utiliza para mantener un seguimiento del árbol de archivos.

```
proc_register_dynamic(struct proc_dir_entry \
    *where, struct proc_dir_entry *self);
proc_unregister(struct proc_dir_entry *where, \
    int inode);
```

así es como se declara un proc root y donde se muestra el directorio del archivo desde la raíz este a su vez se declara dentro del proceso.

La parte de eructo este a su vez reduce tres líneas dentro del módulo inicial y tres dentro del módulo de limpieza, todo lo demás es enviado a la interfaz VFS y está impulsado por eventos.

Las tres líneas en el módulo ini () se parecen a:

```
proc_register_dynamic(&proc_root, \
    &burp_proc_root);
```

y los que están en el módulo de limpieza () parecen:

```
proc_unregister(&proc_root, \
    burp_proc_root.low_ino);
```

El campo ONI bajo es el número de i-nodo del archivo en proceso.

/proc/root está destinado a ser un dispositivo de bloqueo. Depende del modo es el número de bits IFBLK de S, este tiene que tener el mismo que el dispositivo montado desde raíz.

/proc/inmod es un archivo grabable. Necesita sus propias operaciones de archivo para declarar su método de "escritura". Por lo tanto, declara sus operaciones de i-nodo que apuntan a sus operaciones de archivo. Siempre que se llama a su implementación Write (), el archivo pide a kerneld que cargue o descargue el módulo cuyo nombre ha sido escrito.

