

SEP

SECRETARÍA DE  
EDUCACIÓN PÚBLICA



**TECNOLÓGICO NACIONAL DE MÉXICO**  
**Instituto Tecnológico de Pabellón de Arteaga**



**INSTITUTO TECNOLÓGICO<sup>®</sup>**  
**de Pabellón de Arteaga**

**ITEC**

## **Sistemas Operativos I**

### **Lectura – Unidad V**

**Presenta:**

**Sidney Ricardo García Rodríguez.**

**Docente: Eduardo Flores Gallegos**

**INSTITUTO TECNOLÓGICO<sup>®</sup>**  
**de Pabellón de Arteaga**

**ITEC**

## Sistema de archivos virtual en Linux

Uno de los elementos principales en un sistema UNIX es el archivo, y la forma en cómo identificarlo mediante la ruta. Cada archivo se parece en la forma en cómo se accede y modifica. La abstracción de la información se obtiene en Linux a través de la implementación VFS. En los sistemas de archivos de Linux como en UNIX se adoptan conceptos como lo son el superbloque (Es el primer bloque de datos de un disco o partición que contiene metainformación sobre la partición en sí), inode (nodo índice, contiene toda la información sobre un archivo nombrado, excepto su nombre), directorio (Es un archivo que asocia inodos a nombres de archivos) y archivo (Esta asociada con un inode, son áreas de datos pero que también pueden ser directorios).

El sistema operativo debe ser capaz de manejar diferentes formas de distribución de información en el disco, los usuarios cuentan con la necesidad de acceder a todos sus discos duros sin formatear o montar volúmenes. El problema de manejar tantos formatos de archivos se ha abordado mediante los conceptos antes mencionados. Los objetos son un conjunto de operaciones que se usan para manejarlo. Cada objeto incluye una estructura que declara sus propias operaciones y la mayoría reciben un puntero como un primero argumento.

La gestión de datos y el almacenamiento realizado por el kernel en Linux es independiente del formato real de los datos. Entonces el tipo de sistemas de archivos es el módulo de software encargado de mapear las operaciones al mecanismo de almacenamiento real. Este tipo de módulos pueden estar vinculados al kernel. Linux actualmente permite el uso de módulos cargables para todos los tipos de sistemas de archivos, pero es root.

Struct `file_system_type` es una estructura que declara su propio nombre y una función `read_super`. Esta función pasa la información sobre un medio de almacenamiento montado y le pide que rellene la estructura una estructura superbloque, también se puede indicar el tipo de sistema de archivos.

### Listado 1

El kernel usa la estructura de `super_operation` para poder leer y escribir inodos, escribir información de superbloques en el disco y recopilar estadísticas, las `super_operations` declaradas se muestran en el listado 2.

### Listado 2

Después de crearse una copia de memoria del inode, el kernel trabajara sobre ellas usando Struct `inode_operations` que es un conjunto de operación declarado por un módulo de sistemas de archivos. Las operaciones con los inodos tienen la ventaja de que no solo tienen sentido para los directorios, sino que también harán su propia comprobación de errores.

### Listado 3

Las operaciones de archivo especifican como se manejan los datos en archivo real, implementan operaciones de bajo nivel como lectura, escritura, `isearch` y otras. Es interesante observar que la estructura que se plantea aquí, se plantea en los núcleos 2.0 y los 2.1 modifico los prototipos de lectura, escritura e `isearch`.

### Listado 4

No todos los tipos de sistemas de archivos admiten todas las funciones que se describen, ya que no todos cuentan con el concepto inode, aunque cada núcleo identifique cada archivo, si los datos físicos no cuentan



con un inode físico el código que implementa `readdir` y `read_inode` deberá incluir un número de inode para cada archivo. Esta técnica es usar el desplazamiento de bloque de control para el archivo dentro del área de datos del sistema. El sistema de archivos `proc` no tiene un dispositivo físico del cual extraer los datos y por lo cual usa números cableados para archivos que existen.

Un problema típico es el de implementar un sistema de archivos y lidiar con las limitaciones de almacenamiento reales.

#### Listado 5

Existe una parte síncrona del `struct` que reduce a 3 líneas dentro de `init_module` y 3 dentro de `cleanup_module`. Todo esto es enviado por la interfaz VFS ya que es considerado como evento el proceso de acceso a un archivo.

#### Listado 6

Aquí se muestra como aparecen los archivos y cómo funcionan 2 de ellos, en el listado 7 se muestran las estructuras usadas para declarar las entradas de archivos en `/proc`. Estas no se han definido completamente ya que el compilador de C llena con 0 cualquier estructura parcialmente definida sin emitir advertencia alguna.

#### Listado 7

El módulo fue compilado y ejecutado en una PC, Alpha y un Sparc, todos ejecutando Linux en su versión 2.0.x

Esta implementación (`/proc`) tiene características interesantes como lo es la interfaz `sysctl`.