**The “Virtual File System” in Linux**

El elemento de datos principal en cualquier sistema tipo Unix es el "archivo", y un nombre de ruta único identifica cada archivo dentro de un sistema en ejecución como cualquier otro archivo en la forma en que se accede y se modifica: las mismas llamadas al sistema y los mismos comandos de usuario se aplican a cada archivo. Esto se aplica independientemente del medio físico que contiene información y de la forma en que se presenta la información en el medio. Linux analiza su sistema de archivos de la misma forma que Unix 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 si bien se asume un lector este concepto es de que un sistema de archivos realice conceptos de super bloqueo y un directorio de archivos un super bloqueo bloquea datos de disco de partición, un super bloque su concepto de bloqueo de datos que contiene datos de un sistema de archivos como también está el inodo este se asocia con cada archivo es el contiene toda la información sobre un archivo. Un inodo se describe en el núcleo mediante un inodo de estructura el kernel no tiene una estructura de datos especial para representar un directorio cada tipo de sistemas de archivos se utiliza para leer y modificar archivos como los directorios independientes del diseño de sus datos , Por lo general, los archivos son áreas de datos, pero también pueden ser directorios, dispositivos, fijos (primeros en entrar, primero en salir) o en los sockets.

Orientación a Objetos

un sistema operativo debe ser capaz de manejar diferentes formas de distribución de información en el disco. Para esto es óptimo tener información en discos y las particiones del disco los usuario de computadoras necesitan acceder a sus discos sin formatear poner los volúmenes en NFC en la red , el problema de manejar diferentes formatos de datos de una manera transparente se ha abordado haciendo super bloques, inodos y archivos en "objetos"; un objeto declara un conjunto de operaciones que deben usarse para manejarlo, el tipo de sistema de archivos, entonces, es el módulo de software que se encarga de mapear las operaciones al mecanismo de almacenamiento real, ya sea un dispositivo de bloque, una conexión de red (NFS) o prácticamente cualquier otro medio de almacenamiento y recuperación de datos. Estos módulos están vinculados al kernel que se está iniciando o compilado como módulos cargables El kernel usa la estructura para leer y escribir inodos sobre escribir información de superblock en el disco y recopilar estadísticas Después de que se haya creado una copia de memoria del inodo, el kernel actuará sobre ella utilizando sus propias operaciones Las operaciones de manejo de directorios son parte de las operaciones del inodo porque la implementación de una estructura de operaciones traerá condicionales adicionales en el acceso al sistema de archivos. Los problemas de sistemas de archivos de lo anterior es que están diseñados para tener en cuenta toda la semántica de Unix en lo que respecta a los sistemas de archivos. no todos los tipos de sistema de archivos admiten todas las funciones que acabamos de describir, en particular, no todos los tipos tienen el concepto de "inode", aunque el núcleo identifica cada archivo mediante su número de inodo largo sin signo. Ahora me gustaría mostrar un pequeño código para jugar con VFS, pero es bastante difícil concebir un tipo de sistema de archivos.

  proc\_register\_dynamic (struct proc\_dir\_entry \

      \* donde, struct proc\_dir\_entry \* self);

   proc\_unregister (struct proc\_dir\_entry \* donde, \

       int inode);