Sistemas Operativos

ICC329-1

Profesor: Carlos E. Méndez Dumestre
carlos.mendez@ufrontera.cl

Breve Historia

En abril de 1991, Linus Torvalds, de 21 años, empezó a trabajar en unas simples ideas para un núcleo de sistema operativo. Comenzó con un intento por obtener un sistema operativo gratuito similar a Unix que funcionara con microprocesadores Intel 80386. Luego, el 25 de agosto de 1991, Torvalds escribió en el grupo de noticias comp.os.minix:

"*Estoy haciendo un sistema operativo (gratuito, sólo un hobby, no será nada grande ni profesional como GNU) para clones AT 386(486). Llevo en ello desde abril y está empezando a estar listo. Me gustaría saber su opinión sobre las cosas que les gustan o disgustan en minix. Actualmente he portado bash(1.08) y gcc(1.40), y parece que las cosas funcionan. Esto implica que tendré algo práctico dentro de unos meses...*"

El 14 de marzo de 1994, se lanzó ***Linux 1.0.0*** con 176.250 líneas de código.
En marzo de 1995 se lanzó ***Linux 1.2.0*** con 310.950 líneas de código.
El 17 de diciembre de 2003 se lanzó ***Linux 2.6.0*** con 5.929.913 líneas de código.
El año 2012 se liberó el ***Linux 3.x*** con alrededor de 15 millones de líneas de código.
En Junio de 2015 se liberó ***Linux 4.2*** con del orden de los 20 millones de líneas de código.

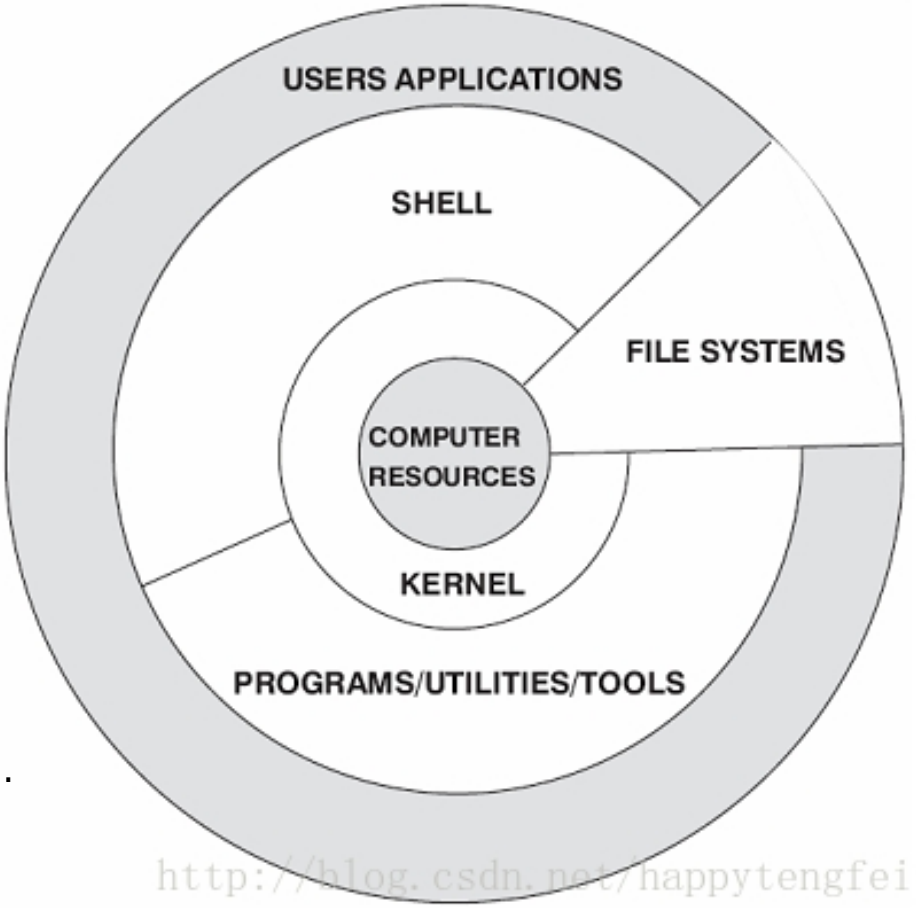
En mayo de 1996 Torvalds decidió adoptar al pingüino Tux como mascota para Linux.

https://www.kernel.org/category/releases.html
https://github.com/torvalds/linux

Arquitectura de GNU/Linux

GNU/Linux cuenta con una arquitectura por capas, donde el Kernel se ejecuta en un espacio privilegiado, muy cerca del hardware, llamado Ring0.

Entorno de escritorio	• GNOME - KDE
Gestor de ventanas	• BlackBox - FluxBox
Sistema de ventanas	• X-Window System
Intérprete de órdenes (Shell)	• Bash - Sh - Csh
Sistema de Archivos	• Ext (2,3,4) - Swap
Kernel	• Linux - Hurd
Hardware	



Según esta estructura, se pueden diferenciar 4 ***componentes*** principales:

1. El Núcleo (**Kernel**)
2. El Intérprete de comandos (**Shell**).
3. ***Herramientas***/Programas del sistema.
4. El **Sistema de Archivos**.

El Kernel de GNU/Linux

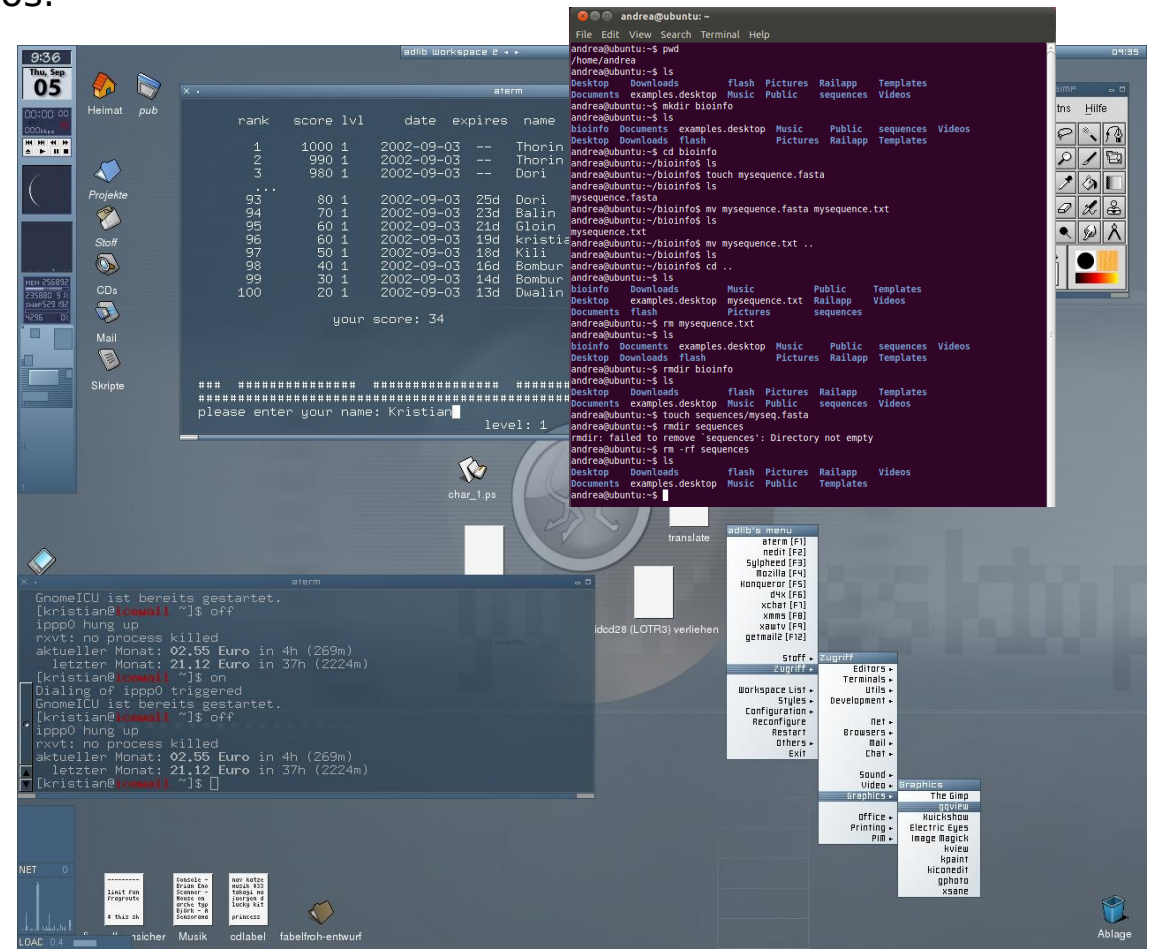
Actualmente Linux es un núcleo ***monolítico híbrido:*** Tiene algunas características de la ***arquitectura monolítica*** pero incorpora además características de la ***arquitectura de capas*** o anillos.
Los controladores de dispositivos y las extensiones del núcleo normalmente se ejecutan en un espacio privilegiado conocido como anillo 0 (***ring 0***), con acceso ***irrestricto*** al ***hardware***, aunque algunos se ejecutan en espacio de usuario.
A diferencia de los núcleos monolíticos tradicionales, los controladores de dispositivos (***drivers***) ***y las extensiones al sistema operativo se pueden cargar y descargar fácilmente como módulos***, mientras el sistema continúa funcionando sin interrupciones.
También, a diferencia de los núcleos monolíticos tradicionales, los controladores pueden ser prevolcados (detenidos momentáneamente por actividades más importantes) bajo ciertas condiciones. Esta habilidad fue agregada para gestionar correctamente interrupciones de hardware, y para mejorar el soporte de Multiprocesamiento Simétrico.

Funciones del Kernel
Administración de Memoria
Administración de Procesos
Control de acceso a periféricos (E/S)

El Shell de GNU/Linux

Proporciona una ***interfaz*** para el usuario. Recibe órdenes del usuario y las envía al ***kernel*** para ser ejecutadas.
El shell proporciona una interfaz entre el kernel y el usuario. Se puede describir como un intérprete: interpreta las órdenes que introduce el usuario y las envía al núcleo. La interfaz del shell es muy sencilla. Normalmente consiste en un inductor desde el que se teclea una orden y después se pulsa enter. En cierta forma, se está tecleando una orden en una línea. A menudo, esta línea se conoce como la Línea de Comandos.

Además de la interfaz de línea de comandos, Linux proporciona una ***interfaz gráfica*** de usuario (***GUI***) llamada ***X-Windows*** que cuenta con varios administradores de ventanas , también llamados ***Entornos de Escritorio***, que puede utilizar. Un administrador de ventana trabaja de forma muy parecida a los GUI de Windows y del Mac, posee ventanas iconos y menús, todos ellos gestionados por medio del ratón. Entre los ***Entornos de Escritorio*** más populares están ***GNOME, KDE***, ***XFCE***, ***Fluxbox*** y el Open Look Window Manager.



Herramientas/Programas de GNU/Linux

Linux contiene un gran número de utilidades. Algunas efectúan operaciones sencillas: otras son programas complejos con sus propios juegos de órdenes. Para empezar, muchas utilidades de pueden clasificar en tres amplias categorías: ***editores***, ***filtros*** y ***programas de comunicaciones***.
También hay utilidades que efectúan ***operaciones con archivos*** y ***administración de programas***.

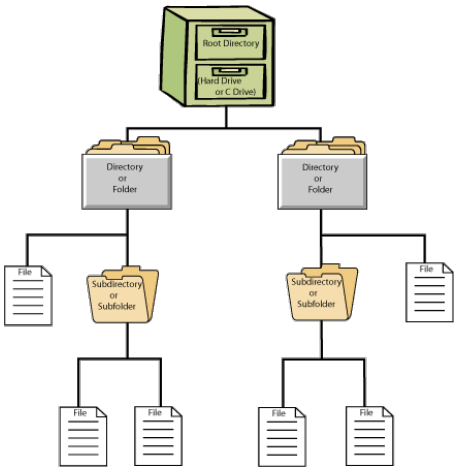
El Sistema de Archivos de GNU/Linux

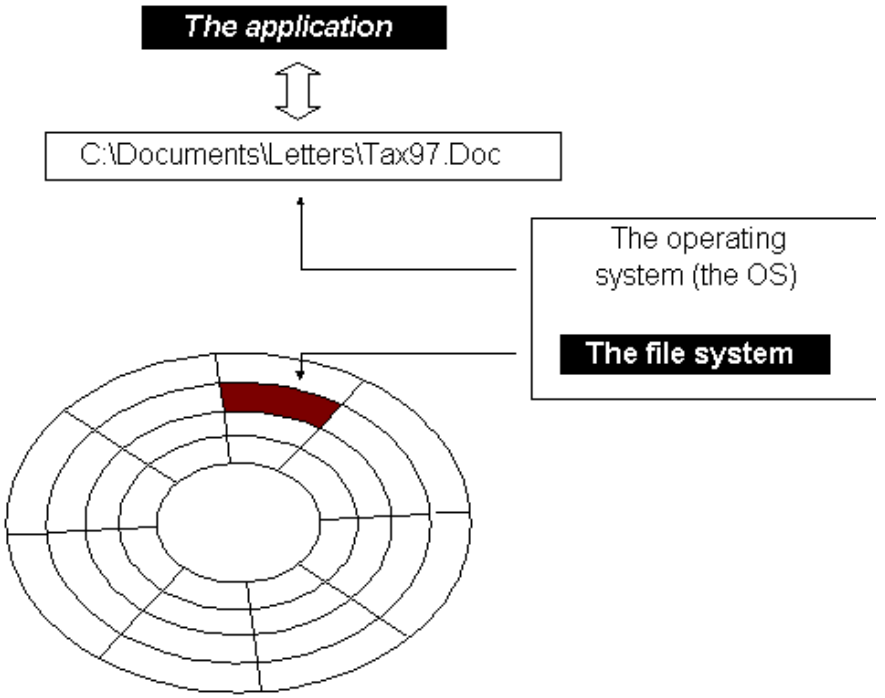
"...En Unix (y en Linux) todo es un archivo..."

¿Qué son los Sistemas de Archivos?

Los ***discos*** son medios de ***almacenamiento***, los cuales pueden contener un S***istema de Archivos***.
Un Sistema de Archivos puede ser comparado con habilitar una biblioteca, se debe primero instalar las estanterías, el sistema de catálogo antes de comenzar a poner los libros en su lugar, de esta forma podremos manejar un orden y saber dónde está cada libro cuando necesitemos ir a leerlo.
Cuando un disco se ***formatea*** éste se organiza y se prepara para recibir datos, es decir, recibe un Sistema de Archivos.
Como ya hemos visto, el Sistema de Archivos es una parte integral del Sistema Operativo y a veces un Sistema Operativo es capaz de trabajar con más de un Sistema de Archivos (como el caso de Windows y Linux).
Algunos sistemas de archivos conocidos: ***FAT16/FAT32 (Ms-DOS y Windows), NTFS (Windows), UFS (Unix), HPFS (MacOS), EXT (Linux).***

El ***Sistema de Archivos*** conoce dónde estan almacenados los archivos, encuentra y lee los sectores importantes y entrega esta información al ***Sistema Operativo***.

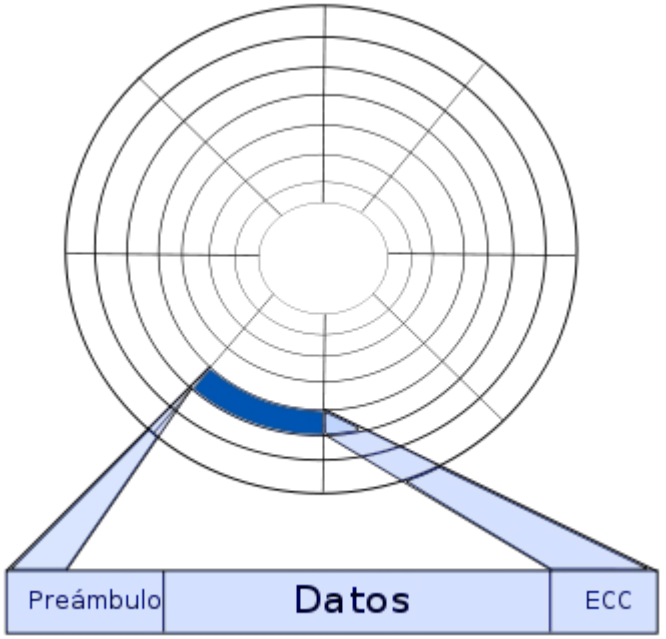




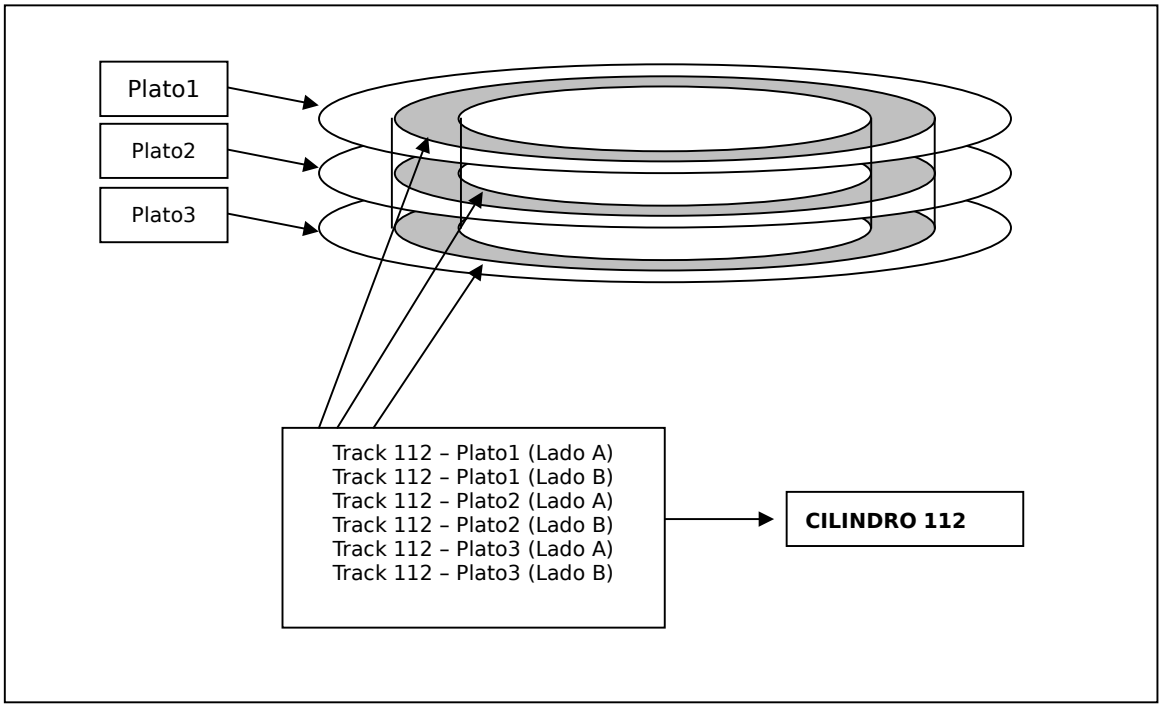
El Formato

Los ***discos*** deben ser ***formateados*** para que puedan almacenar los datos. Existen dos tipos de formato:

1. Formato de Bajo Nivel



También llamado formato físico, es realizado por software y consiste en colocar “marcas” en la superficie de óxido metálico magnetizable de Cromo o Níquel,2 para dividirlo en ***pistas*** concéntricas y estas, a su vez, en sectores los cuales pueden ser luego referenciados indicando la cabeza lectora , el ***sector*** y ***cilindro*** que se desea leer. El tamaño estándar de cada sector es de ***512 bytes***.



2. Formato de Alto Nivel

El ***formato lógico***, de alto nivel o ***también llamado sistema de archivos***, puede ser realizado habitualmente por los usuarios, aunque muchos medios vienen ya formateados de fábrica. El formato lógico implanta un sistema de archivos que asigna sectores a archivos. En los discos duros, para que puedan convivir distintos sistemas de archivos, antes de realizar un formato lógico hay que dividir el disco en ***particiones***; más tarde, cada partición se formatea por separado.

Cada sistema operativo tiene unos sistemas de archivos más habituales:

- ***Windows***: FAT, FAT16, FAT32, NTFS, EFS, ExFAT.
- ***Linux***: ext2, ext3, ext4, JFS, ReiserFS, Reiser4, XFS.
- ***Solaris***: UFS, ZFS.
- ***Mac OS***: HFS, HFS+.
- ***IBM***: JFS, GPFS.
- ***Discos Opticos***: UDF.

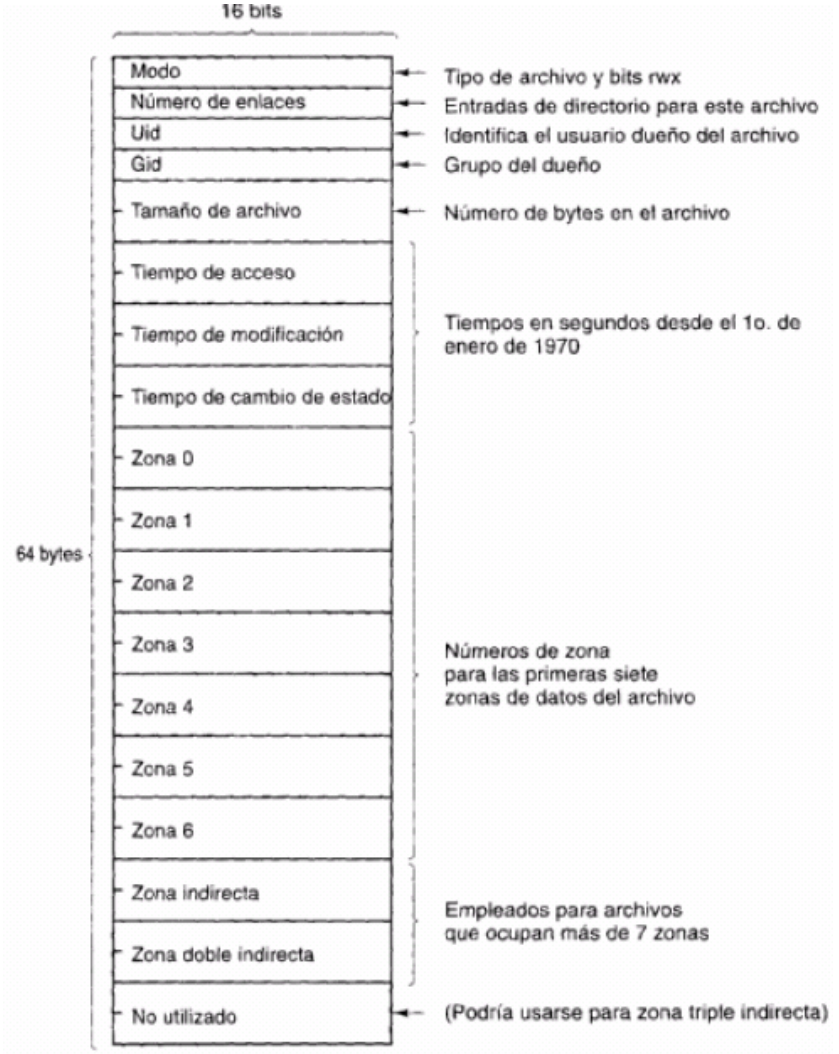
El Sistema de Archivos EXT

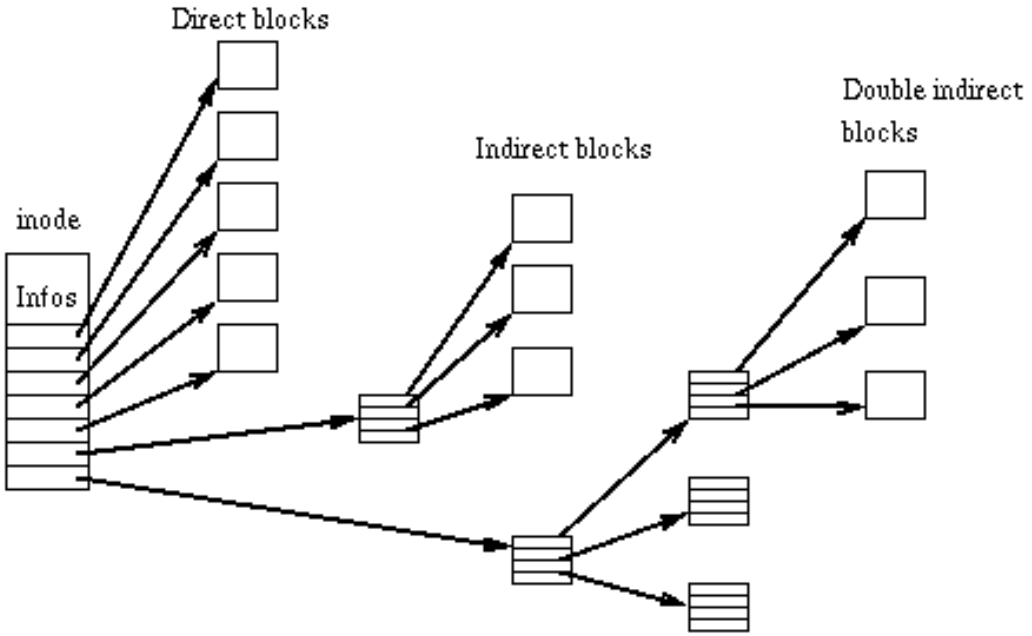
El sistema ***EXT (Extended File System)*** es el Sistema de Achivos utilizado por Linux. Actualmente Linux cuenta con la cuarta versión de dicho Sistema de Archivos, la EXT4.
EXT nace como una versión mejorada del sistema de archivos de MINIX, incorpora mejoras a nivel de rendimiento y de integridad de los datos.
En EXT encontramos incorporado el concepto de ***JOURNALING****:*
Un sistema de archivos Journaling, se puede definir como un sistema que sirve para darles más seguridad a la integridad de los datos, y los metadatos, que se encuentran en el disco duro.
Estos sistemas de archivos se recomiendan para ***sistemas de alta disponibilidad***. Una de las principales características que poseen es que ***registran todos los cambios (transacciones) por realizarse en un historial (Journal) antes de ejecutar dichos cambios directamente en el sistema***. Por esta razón un sistema de archivos Journaling ***tiene la posibilidad de "volver atrás"*** en caso de que se detecten errores, caídas del sistema, etc. Y por lo tanto son menos corruptibles en el tiempo. (Por ejemplo, en el caso de una caída del sistema operativo por una falla de energía, las transacciones al disco que no fueron completadas, fueron registradas anteriormente en el "journal", y al reiniciar la máquina éstas son realizadas, por lo tanto el sistema de archivos se "sincroniza" de vuelta sin perder datos. En otros sistemas de archivos, si sucediera esto, se podían perder muchos datos.)

Componentes de EXT
Inodos
Directorios
Links
Archivos de Dispositivos
Virtual File System

Inodos

En ***EXT***, cada archivo está representado por una estructura llamada ***INODO***.
Cada inodo contiene la ***descripción del archivo***: tipo de archivo, los privilegios de acceso , propietarios , marcas de tiempo , tamaño, punteros a bloques de datos.
Las direcciones de los bloques de datos asignados a un archivo se almacenan en su inodo.

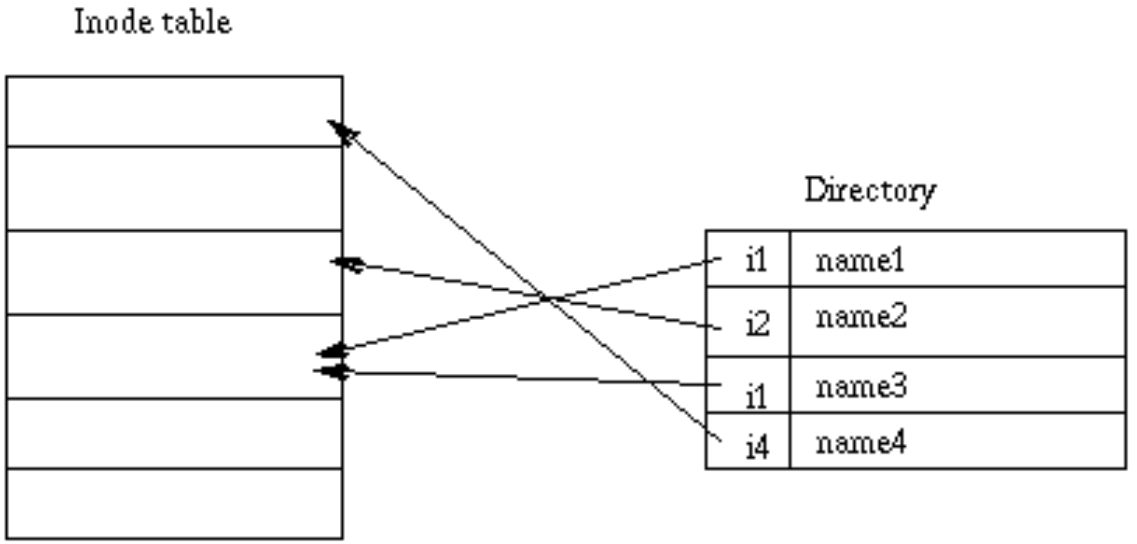




Un ***Bloque*** corresponde a un conjunto de ***sectores de disco*** que puede ir entre 1KB y 64KB (2 a 128 sectores de disco).

Directorios

Los ***Directorios*** están estructurados en un ***árbol jerárquico***. Cada directorio puede contener archivos u otros directorios.
Los Directorios están implementados como un tipo especial de archivo. En realidad , ***un directorio es un archivo que contiene una lista de entradas***. Cada entrada contiene un número de inodo y un nombre de archivo. Cuando un proceso utiliza una ruta de archivo,el kernel búsca en los directorios para encontrar el ***número de inodo*** correspondiente. Después de que el nombre se ha convertido en un número de inodo, el inodo se carga en memoria y es utilizado por las solicitudes posteriores.



Links

Sistemas de archivos EXT implementan el concepto de ***LINK*** (enlace). Varios nombres se pueden asociar con un inodo. El inodo contiene un campo que contiene el número asociado con el archivo. ***Agregar un enlace simplemente consiste en la creación de una entrada de directorio***, donde el número de inodo apunta al inodo, y en incrementar la cuenta de vínculos en el inodo. Cuando se ***elimina*** un enlace, es decir, cuando se utiliza el comando ***rm*** para eliminar un nombre de archivo, ***el Kernel disminuye la cuenta de los enlaces y desasigna el inodo si este recuento se convierte en cero.***

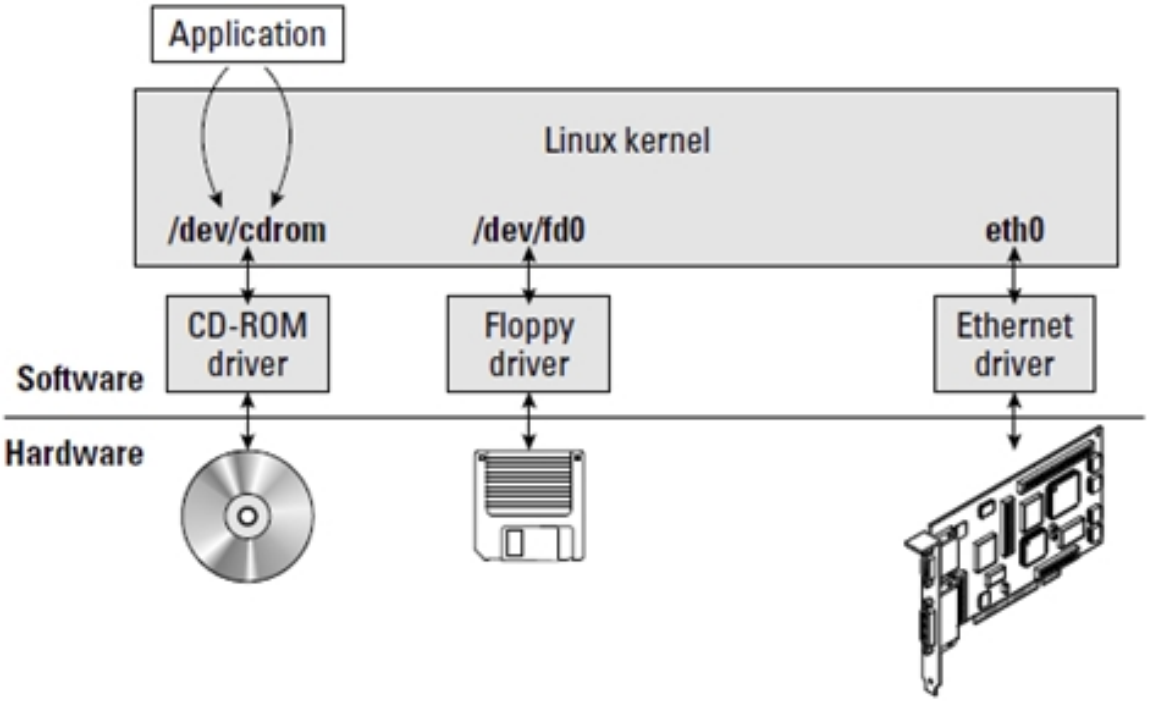
Este tipo de enlace se denomina ***Hard Link*** y sólo se puede utilizar dentro de un único sistema de archivos: ***es imposible crear un hard link que apunte a otro sistema de archivos***. Por otra parte, los ***hard links sólo pueden apuntar a un archivo***. No se permite crear un hard link a un directorio, para evitar la aparición de un ciclo en el árbol de directorios.

Existe otro tipo de Links en la mayoría de los sistemas de archivos EXT: Los ***Symbolic (Soft) Links*** (enlaces simbólicos) son simplemente ***archivos que contienen un nombre de archivo***. Cuando el Kernel se encuentra con un ***Soft Link***, simplemente sustituye el nombre del enlace por su contenido, es decir, el nombre del archivo de destino, y reinicia la interpretación ruta. Debido a que un ***Soft Link no apunta a un inodo***, es posible crear enlaces entre distintos sistemas de archivos. Los enlaces simbólicos pueden apuntar a cualquier tipo de archivo, incluso en archivos inexistentes. Los Soft Links no tienen las limitaciones asociadas a Hard Links. Sin embargo, utilizan un poco de espacio en disco, asignado por su inodo y sus bloques de datos, y causan sobrecarga de trabajo en el acceso a los inodos ya que el kernel tiene que reiniciar la interpretación nombre cada vez que encuentra un enlace simbólico.

Archivos de Dispositivos

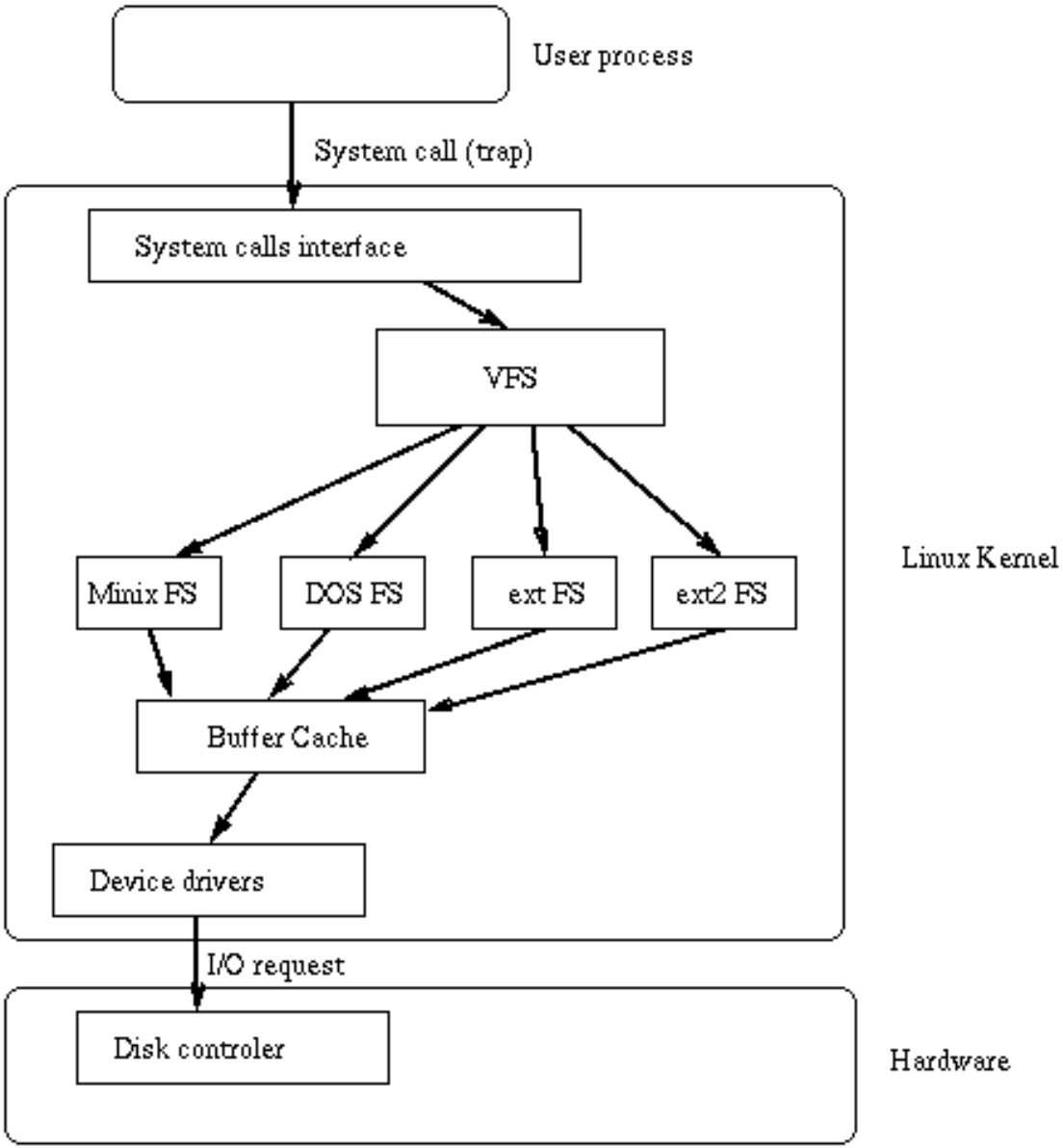
En los ***sistemas operativos tipo Unix***, los ***dispositivos se pueden acceder a través de archivos especiales***. Un archivo especial de dispositivo no utiliza ningún espacio en el sistema de archivos . Es solamente un punto de acceso al controlador de dispositivo .

Existen dos tipos de archivos especiales: ***caracter*** y ***bloque***. La primera permite operaciones de ***E/S en modo de caracteres*** mientras que el segundo requiere que los datos sean ***escritos en modo bloque*** a través del funciones caché de ***búfer***. Cuando una solicitud de E/S se realiza en un archivo especial , se envía a un (pseudo) controlador de dispositivo . Un archivo especial hace referencia a un número, que identifica el tipo de dispositivo, y un número de menor escala, que identifica la unidad.



Virtual File System

El Kernel de Linux contiene una capa de ***Sistema de Archivos Virtual*** que se utiliza durante las llamadas al sistema que actúan en los archivos. La VFS es una capa intermedia que se ocupa de las llamadas al sistema de archivos y llama a las funciones necesarias en el código del sistema de archivos físico para hacer la E/S.
Este mecanismo intemedio se utiliza con frecuencia en los sistemas operativos tipo Unix para facilitar la integración y el uso de varios tipos de sistemas de archivos.
El ***VFS*** define un conjunto de funciones que cada sistema de archivos tiene que poner en práctica. Esta ***interfaz*** se compone de un conjunto de operaciones asociadas a los tres tipos de objetos: los ***sistemas de archivos, los inodos y los archivos***.
El ***VFS*** "sabe" qué tipos de ***sistemas de archivos están soportados en el kernel***. Utiliza una tabla definida durante la configuración del kernel. Cada entrada en esta tabla describe un tipo de sistema de archivos: contiene el nombre del tipo de sistema de archivos y un puntero en una función, que es llamada durante la operación de montaje. Esta función es responsable de leer el superbloque desde el disco, inicializando sus variables internas, y devolver un ***descriptor del sistema de archivos montado*** a la VFS. Después que se monta el sistema de archivos, las funciones VFS pueden utilizar este descriptor para acceder a las rutinas del sistema de archivos físicos.



Característcas de EXT4

File system	Maximum filename length	Allowable characters in directory entries[4]	Maximum pathname length	Maximum file size	Maximum volume size[5]
FAT32	8.3 (255 UTF-16 characters with LFN)[7]	Any Unicode except NUL (with LFN)[7][8]	No limit defined[9]	4 GiB	512 MiB to 8 TiB[12]
HPFS	255 bytes	Any byte except NUL[13]	No limit defined[9]	2 GiB	2 TiB[14]
NTFS	255 characters	Any Unicode except NUL, /	32,767 Unicode characters with each path component (directory or filename) up to 255 characters long[9]	16 EiB[15]	16 EiB[15]
ext2	255 bytes	Any byte except NUL[8]	No limit defined[9]	16 GiB to 2 TiB[5]	2 TiB to 32 TiB
ext3	255 bytes	Any byte except NUL[8]	No limit defined[9]	16 GiB to 2 TiB[5]	2 TiB to 32 TiB
ext4	255 bytes	Any byte except NUL[8]	No limit defined[9]	16 GiB to 16 TiB[5][19]	1 EiB

EiB: ExaByte = 1 Millón de TeraBytes
(https://en.wikipedia.org/wiki/Orders_of_magnitude_(data))

https://en.wikipedia.org/wiki/Comparison_of_file_systems