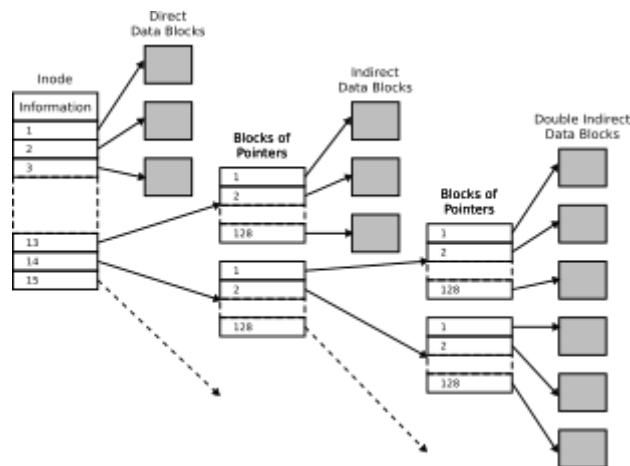


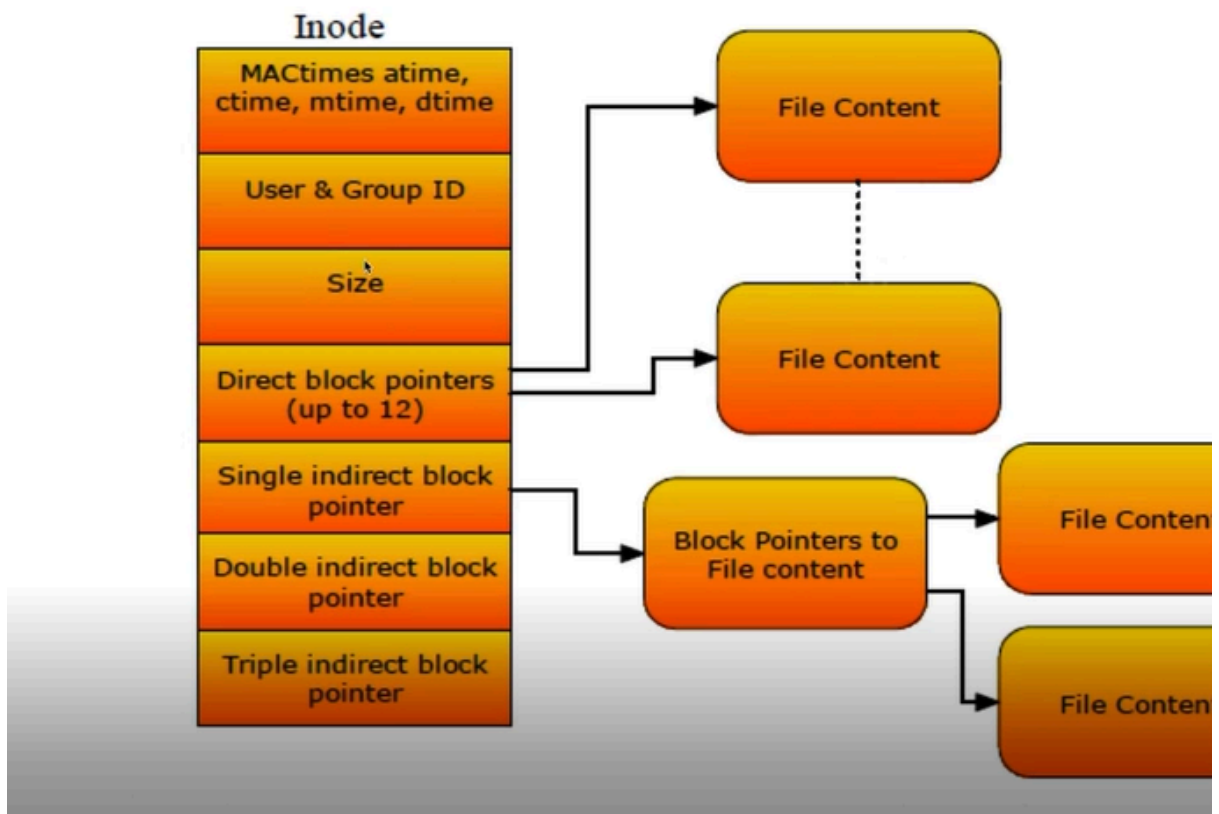
- Hay ficheros muy importantes a la hora de manejar el SO y además también la extensión de un fichero y como declara ficheros ocultos, esto hay que tenerlo bien controlado.
- Que es un fichero? es un conjunto de información relativa bajo un mismo concepto que se guarda bajo un nombre que lo identifica
- en GNU/LINUX el nombre de un fichero puede tener hasta 255 caracteres exceptuando el carácter /, que se utiliza para el directorio raíz y para separar directorios
- cada fichero se identifica además de por su nombre por su número de i-nodo que es un número entero único que identifica cada fichero
- el I-NODO si identifica el fichero único dentro de un sistema de archivos
- Existe en todo los sistemas de ficheros una tabla de asignación de ficheros y cada fichero tiene un número de i-nodo
- por cada i-nodo vamos a guardar su número i-nodo, su tipo de fichero, propietario y grupo, permisos, fecha de creación, acceso y última modificación, enlaces
- ESTRUCTURA: Estructura del inodo

El estándar POSIX establece un modelo de sistema de archivos que se ajusta al empleado en los UNIX tradicionales. Un archivo ordinario tendrá las propiedades siguientes:

- El *identificador de dispositivo* del dispositivo que alberga al sistema de archivos.
- El *número de inodo* que identifica al archivo dentro del sistema de archivos
- La *longitud* del archivo en bytes.
- El *identificador de usuario* del creador o un propietario del archivo con derechos diferenciados
- El *identificador de grupo* de un grupo de usuarios con derechos diferenciados
- El *modo* de acceso: capacidad de leer, escribir, y ejecutar el archivo por parte del propietario, del grupo y de otros usuarios.
- Las *marcas de tiempo* con las fechas de última modificación (*mtime*), acceso (*atime*) y de alteración del propio inodo (*ctime*).
- El *número de enlaces*, esto es, el número de nombres (entradas de directorio) asociados con este inodo. El número de enlaces se emplea por el sistema operativo para eliminar el archivo del sistema de ficheros, tanto el inodo como el contenido, cuando se han borrado todos los enlaces y el contador queda a cero.
- La *estructura de punteros*, para direccionar hacia los bloques de datos (contenido) del archivo. Está compuesta por:
 - Doce punteros que apuntan directamente a bloques de datos del archivo (punteros directos)
 - Un puntero de indirección simple (apunta a un bloque de punteros, los cuales apuntan a bloques de datos del archivo)
 - Un puntero de indirección doble (apunta a un bloque de punteros, los cuales apuntan a otros bloques de punteros, estos últimos apuntan a bloques de datos del archivo)
 - Un puntero de indirección triple (apunta a un bloque de punteros que apuntan a otros bloques de punteros que apuntan a otros bloques de punteros que luego apuntan a bloques de datos del archivo)



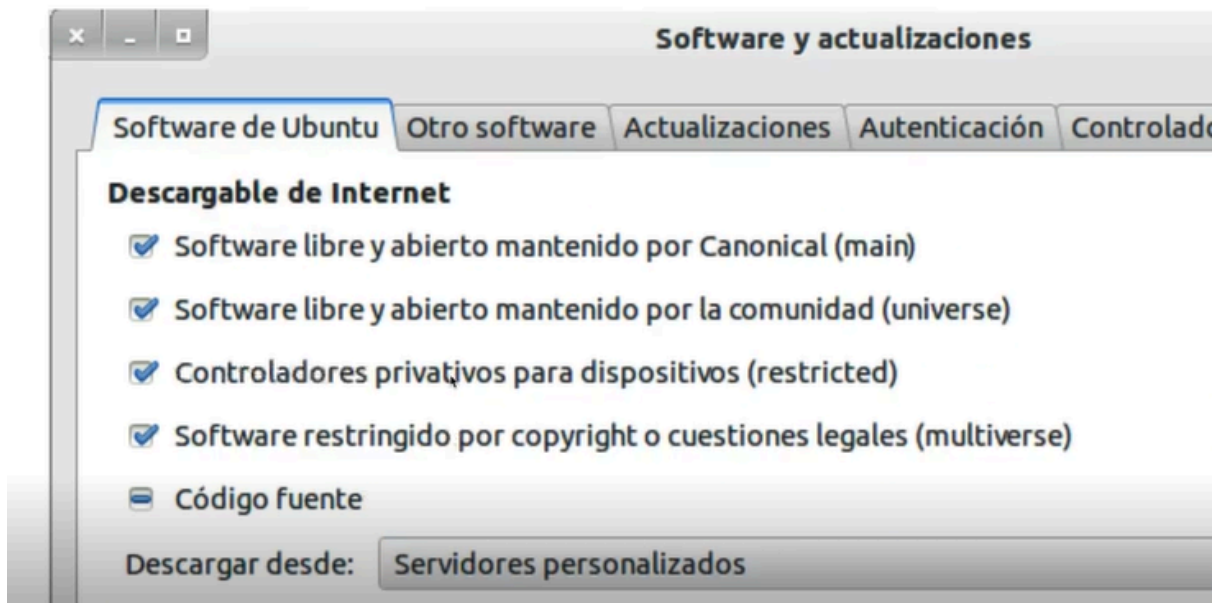
○ Enlaces



- cuando un fichero es muy grande, el sistema fragmenta la información en otros ficheros
- cada vez que modificamos algún atributo del fichero lo que se modifica es el i-nodo de dicho fichero
- Operaciones sobre los archivos: Las operaciones que se pueden realizar sobre los ficheros son las siguientes: Crear, borrar, copiar, mover (mueve el fichero original al destino que le indiquemos o lo renombra), modificar, actualizar fichero, seleccionar, renombrar, abrir, cerrar, enlazar (crear un vínculo al fichero)
- de los ficheros podemos distinguir entre un fichero regular, un enlace simbólico y enlace duro.
- Directorios
- Regulares: información de cualquier tipo

- enlaces: única información que guarda a qué fichero enlaza, pero hay 2 tipos, enlaces simbólicos y absolutos o llamados enlaces blandos y enlaces duros
- en terminal ejecutamos `ls` nos muestra la lista de ficheros que hay en el directorio el
- `pwd` nos muestra el directorio en el que estamos actualmente
- con `ls -l` nos muestra la lista con metadatos
- `ls -li` nos muestra el número de i-nodo de cada uno de los ficheros
- enlace duro consiste en crear otro fichero con otro nombre distinto pero que tiene el mismo número de i-nodo, si creamos un enlace duro a un fichero los cambios realizados en el fichero o enlace se reflejan en el origen y viceversa, es como si el fichero tuviera dos nombres, si se borra uno de los dos, el otro sigue existiendo (digamos que con el enlace duro hay otros archivos que apuntan a la misma zona del disco duro, pero si modificamos algunos de los 2 estamos modificando la misma zona del disco duro, o sea que se refleja)
- enlace simbólico: es un acceso directo a un fichero desde otra ubicación, cada fichero tiene su propio número de i-nodo, en el caso de que se borre el original, en enlace simbólico perdería (2 ficheros que apuntan a la RUTA donde está el otro archivo, son dos cosas muy diferentes) sería como un acceso directo de windows, si cambio la ruta ya el enlace simbólico ya no funciona porque solo guarda la ruta, en cambio en enlace duro apunta a la zona de disco donde está el archivo
- `ln` necesita 2 argumentos, fichero original y un enlace: ej `ln prueba.txt duro.txt` y ahora hacemos un `ls -l` y veremos 2 archivos uno llamado `prueba.txt` y el otro llamado `duro.txt`, metemos `ls -li` y vemos que el `duro.txt` tiene un número de i-nodo o sea que ambos apuntan a la misma zona de disco, entonces si ahora modifico `duro.txt` y guardo y luego abro `prueba.txt` está modificado, porque accedo a la zona de disco de enlaces duros. pero si borramos 1 el otro queda intacto, porque la zona de disco estaba apuntada por 2 y ahora por 1
- para enlace simbólico es: `ln -s prueba.txt mendoza.txt` y luego hacemos `ls -l` y vemos que `mendoza.txt` me dice a qué ruta está apuntando y si listamos los inodos con `ls -li` y vemos que el número no tiene nada que ver y si abrimos `mendoza.txt` vemos que apunta a `duro.txt`, y si renombramos `duro.txt` y cambiamos el nombre de `duro.txt` ahora el enlace blando falla porque apunta mal ya no existe con ese nombre
- `ls -a` para ver ficheros ocultos
- en linux todo fichero que comience con `.` para el sistema es fichero oculto
- Shell Scripts: en linux tenemos unos archivos llamados shell scripts o solo scripts que contienen una serie de comandos que se irán ejecutando uno detrás de otro en la shell, terminal o consola. Es el equivalente a un fichero `.bat` en MS WINDOWS. Estos ficheros pueden estar creados por el sistema operativo o por un usuario para mecanizar
- ¿Qué es la extensión de un fichero? simplemente es una parte del nombre del fichero pero no determina la naturaleza del fichero, lo único que hace es como ponerle un nombre y la única función que tiene es decirle al SO con qué programa tienes que abrirlo
- Nos permiten definir ciertos parámetros a la hora de arrancar un sistema
- permiten elegir los programas que se cargaran al inicio del sistema
- los principales ficheros son `etc/grub.d` y `etc/default/grub`, esto guarda la configuración del gestor de arranque grub, es decir aquí se están los parámetros y opciones necesarios para arrancar los distintos SO (2 o más sistemas operativos)
- `etc/fstab`: nos indica qué particiones quiere que arranquen

- `etc/apt/sources.list`: repositorios de programas que queremos instalar
- parte grafica:



- `etc/hostname`: contiene el nombre de nuestra máquina
- `etc/profile` y `etc/profile.d`: guarda información de programas de arranque del sistema
- `etc/network/interfaces`: contiene información sobre la configuración de la red y de las tarjetas de red. Con el comando `IFCONFIG` podemos ver la configuración
- comando para imprimir contenido de archivos `CAT` + ruta
- `etc/crontab`: contiene información sobre tareas o procesos programados
- `etc/passwd`: contiene información sobre todos los usuarios que existen en el sistema indicando la siguiente información:
 - nombre, una x que antiguamente almacenaba la clave del usuario
 - UID identificador unico de usuario
 - GID identificador unido de grupo que pertenece el usuario
 - Comentario: texto
 - Directorio HOME, directorio de trabajo del usuario
 - Shell que conjunto de comandos puede utilizar el usuario
- por defecto cuando instalamos un programa linux crear usuarios que no sirven para iniciar sesion pero es como un usuario que sirve para iniciar el programa, porque cuando ejecutemos el comando veremos muchos usuarios
- `etc/group`: información de todos los grupos que hay en el SO
 - Nombre del grupo
 - una X que indica que su clave de grupo (si la hay)
 - GID identificador único de grupo
 - lista de usuarios que pertenecen al grupo
- `etc/shadow`: contiene las contraseñas encriptadas
- `etc/login.defs`: opciones de configuración por defecto que tiene el usuario