

1. INTRODUCCIÓN A GNU/LINUX Y EL SOFTWARE LIBRE.

GNU/Linux es un sistema operativo, para empezar. Significa esto que es un conjunto de instrucciones que nos va a permitir "hacer cosas", cualquier tipo de tarea, con nuestro ordenador. Sus características más notables son:

- Proviene de UNIX, otro sistema operativo, y es casi un derivado de él.
- Es parte del proyecto GNU, lo que significa, entre otras muchas cosas, que es libre y que muchas veces no estás obligado a pagar por él. Puedes usarlo de modo gratuito y además puedes modificar su código fuente, para adaptarlo a tus propias necesidades o para contribuir en su continuo desarrollo, en el que toman parte programadores de todo el mundo (tú puedes ser uno de ellos). Antes de seguir leyendo y aprendiendo sobre él, conviene que visites la página web del proyecto GNU (<http://www.gnu.org>) en la que se dan algunos detalles sobre qué es el proyecto, qué puedes hacer y qué no puedes hacer con el código fuente además de algunos otros detalles importantes.
- Es potente, es seguro y es estable. Es por esto por lo que muchas veces es considerado un sistema operativo sólo para servidores, cumple muy bien esta función. Pero como sistema operativo de oficina, de escritorio o de publicación es también excelente.
- Infinidad de programas para cualquier tarea. Muchos de ellos son gratuitos (no por ello menos calidad que los comerciales, pues muchas veces los superan). También existen aplicaciones comerciales.
- Es un sistema multiusuario real y multitarea, funciona de manera muy productiva en redes. Siempre lo ha sido, desde sus raíces UNIX. En definitiva Linux pone todo lo bueno de los grandes UNIX y algunas cosas más al alcance de todo el mundo. ¿Lo vas a dejar ahí? ¿Lo vas a desaprovechar? Un consejo, yo no lo haría.

Hablemos un poco de la historia de GNU/Linux. Aquí tenemos a un personaje muy importante, que podemos considerar como el creador de Linux. Su nombre es Linus Trovalds, y ahora mismo sigue coordinando el trabajo de los programadores de todo el mundo en el núcleo del sistema operativo para mejorarlo y adaptarlo al nuevo hardware. Puede decirse que Linus Trovalds creó este sistema operativo en 1991.

Escribió un KERNEL (parte principal de un sistema operativo) y lo dejó a disposición de sus amigos y luego de toda la Internet para que cualquiera pudiese mejorarlo. Siempre se ha seguido la regla POSIX para que el sistema operativo sea compatible con otros UNIX y de esta manera tener mayor software a su disposición, y que el que se cree para Linux valga también en otros UNIX. Algún tiempo más tarde, para adaptar el sistema operativo y facilitar la instalación, nacieron las compañías distribuidoras de Linux.

Entre ellas destacan RedHat, Debian, Caldera, SuSE, Mandrake... Estas distribuidoras pueden cobrar por ofrecer sus productos en CD o en algún otro soporte, pero deben poner su código a disposición del público, si han utilizado software GNU (bajo licencia GPL, que dice entre otras cosas que si usas código GPL en tu programa, tu programa se convierte automáticamente en GPL).

¿Cabe la posibilidad de preguntarse cuál es la mejor distribución de Linux? GNU/Linux tiene muchas distribuciones, y por lo general no existe una distribución mejor que las demás, cada una de ellas tiene sus puntos a favor en determinados aspectos. Debian es una distribución que no es realmente comercial en sí como lo son la mayoría de las otras. El desarrollo de Debian es muy parecido al voluntariado que existe en el desarrollo del kernel de Linux. Para los que se aproximan por primera vez a Linux, es cierto que Mandrake, si se tiene una máquina con recursos, ofrece una instalación y configuración sencillas y un entorno gráfico personalizado ideal para no perderse al principio. Claro que depende de tí, si tienes la oportunidad de tratar con varias distribuciones, trabaja con la que te sientas más cómodo.

¿Puede haber problemas? Dependiendo del hardware que tengas en tu ordenador, porque Linux no soporta algunos dispositivos como Winmódems (módems diseñados para funcionar sólo con MS Windows, en su mayoría internos), algunas tarjetas de sonido, etc.

¿Puedo tener Windows y Linux en la misma máquina? ¡Claro! Como veremos más adelante la clave está en particionar el disco duro, para que Windows use una parte y Linux use otra, de manera que no interfieran. También como veremos después, el cargador de arranque de Linux te permitirá elegir qué sistema operativo iniciar.

¿Qué es Debian?

- Sistema operativo libre desarrollado por la comunidad.
- Conjunto de programas básicos y utilidades.
- Utiliza el núcleo Linux (el corazón del sistema operativo).
- Herramientas básicas Proyecto GNU -> GNU/Linux.

2. ESTRUCTURA DEL SISTEMA DE ARCHIVOS

El sistema de archivos es más o menos "la forma de escribir los datos en el disco duro". El sistema de archivos nativo de Linux es el EXT2. Ahora proliferan otros sistemas de archivos con journalising (si se arranca sin haber cerrado el sistema, no necesitan hacer un chequeo sino que recuperan automáticamente su último estado), los más conocidos son EXT3, ReiserFS y XFS.

La estructura de directorios que sigue Linux es parecida a la de cualquier UNIX. No tenemos una "unidad" para cada unidad física de disco o partición como en Windows, sino que todos los discos duros o de red se montan bajo un sistema de directorios en árbol, y algunos de esos directorios enlazan con estas unidades físicas de disco.

MUY IMPORTANTE: Las barras en Linux al igual que en cualquier UNIX son inclinadas hacia la derecha, como se puede ver más abajo (ese es el motivo de que en internet sean inclinadas hacia la derecha ya que nació bajo UNIX y en Linux podremos aprovechar todas sus ventajas). Expliquemos esto más a fondo, incluyendo los directorios principales:

Tabla: Estructura de directorios en Linux

| Directorio | Descripción |
|-------------------|---|
| / | Es la raíz del sistema de directorios. Aquí se monta la partición principal Linux EXT. |
| /etc | Contiene los archivos de configuración de la mayoría de los programas. |
| /home | Contiene los archivos personales de los usuarios. |
| /bin | Contiene comandos básicos y muchos programas. |
| /dev | Contiene archivos simbólicos que representan partes del hardware, tales como discos duros, memoria. |
| /mnt | Contiene subdirectorios donde se montan (se enlaza con) otras particiones de disco duro, CDROMs, etc. |
| /tmp | Ficheros temporales o de recursos de programas. |
| /usr | Programas y librerías instalados con la distribución |
| /usr/local | Programas y librerías instalados por el administrador |
| /sbin | Comandos administrativos |
| /lib | Librerías varias y módulos ("trozos") del kernel |
| /var | Datos varios como archivos de log (registro de actividad) de programas, bases de datos, contenidos del servidor web, copias de seguridad. |
| /proc | Información temporal sobre los procesos del sistema (explicaremos esto más en profundidad posteriormente). |

Nombres de archivos en Linux

Los nombres de archivos en Linux (como en todos los UNIX) distinguen mayúsculas de minúsculas, esto es, son "case sensitive". Los archivos README, readme, REadme y rEadme por ejemplo son archivos distintos y por lo tanto al ser nombres distintos pueden estar en el mismo directorio.

En Linux los archivos no tienen por qué tener una extensión. La suelen tener a modo orientativo, pero no es en absoluto necesario. Linux sabe qué contiene cada archivo independientemente de cuál sea su extensión. Por comodidad, podremos llamar a todos nuestros archivos de texto con la extensión .texto, o a todos nuestros documentos con la extensión .documento, de esta manera, podremos luego agruparlos más fácilmente.

Los ficheros y directorios ocultos en Linux comienzan su nombre por un punto (.). Los nombres de archivos o directorios pueden ser muy largos, de más de 200 caracteres, lo cual nos da bastante flexibilidad para asociar el nombre de un archivo a lo que contiene. No obstante, hay ciertos caracteres que nunca se deberían utilizar a la hora de nombrar un archivo. Uno de ellos es el espacio, nunca llamaremos a un fichero con un nombre que contenga un espacio. Tampoco son recomendados otros caracteres raros como signos de puntuación (a excepción del punto), acentos o la ñ. En algunos casos Linux ni siquiera nos permitirá usarlos. Los recomendables son las letras A-Z, a-z, los números (0-9), el punto, el guión (-) y el guión bajo (_) para nombrar un archivo. Los acentos y la ñ tampoco se recomiendan.

3. DISPOSITIVOS

Para Linux todo es un archivo, incluyendo dispositivos como discos duros, cdroms, disquetes, unidades de cinta, memorias usb, etc., así como dispositivos de comunicación como puertos seriales y paralelos, modems, etc, incluso también las consolas o terminales son dispositivos asociados a un archivo. Estos dispositivos son enlazados (linked) a un dispositivo de archivo, es decir un dispositivo físico es representado o asociado a un archivo. Estos archivos se encuentran bajo el directorio /dev.

Los dispositivos en Linux son identificados con un designador de dos o tres letras, además si el dispositivo admite particiones se utiliza una progresión numérica o alfabética para identificar la partición. En la siguiente tabla se indica el designador de tipo dispositivo y su descripción.

| Tipo | Dispositivo |
|------|--|
| hd | Discos duros IDE y dispositivos IDE como cdroms |
| sd | Discos duros SCSI |
| scd | Cdrom SCSI |
| st | Unidades de cinta SCSI |
| ht | Unidades de cinta IDE |
| fd | Unidades de disquetes |
| lp | Puertos paralelos |
| tty | Terminales o consolas |
| pty | Terminales remotas o de red, incluyendo las abiertas en Window X |
| ttyS | Puertos seriales |
| cua | Puertos de comunicación |
| eth | Tarjetas o interfaces de red ethernet |

Los dispositivos que admiten particiones generalmente estas se designan con letras, así por ejemplo las unidades IDE que son las más comunes en cualquier PC actual su designación sería la siguiente:

| Canal IDE | Dispositivo Linux | Equivalente Windows |
|--------------------|-------------------|---------------------|
| Primario Maestro | /dev/hda | C: |
| Primario Esclavo | /dev/hdb | D: |
| Secundario Maestro | /dev/hdc | E: |
| Secundario Esclavo | /dev/hdd | F: |

4. INSTALACIÓN DEL SISTEMA

Requerimientos de Hardware de Linux

Linux soporta plataformas de hardware tales como Intel x86, PowerPC, S/390, SPARC y Alpha. Los diferentes requerimientos de hardware para Linux se listan en la siguiente tabla:

| Hardware | Requerimientos |
|------------------------------|--|
| CPU | La serie x86 de Intel y sus compatibles, DEC, Alpha, Motorola, PowerPC, etc. |
| Tarjeta Madre (MotherBoard) | Sistemas de bus PCI, EISA, VESA y MCA. |
| Memoria | 64 MB(mínimo), 256 MB recomendados para mayor eficiencia y ejecución |
| Monitor y Adaptador de Video | CGA, EGA, VGA, IBM monochrome, Súper VGA y otras tarjetas aceleradoras de video |
| Dispositivos de Puntero | Ratón serial estándar como Logitech, serie MM, Microsoft 2 botones, Sistemas Mouse de 3 botones, etc. |
| Controlador de Disco Duro | IDE, EIDE, MFM \, RLL y la mayoría de los controladores ESDI |
| Espacio de Disco Duro | Requiere un mínimo de 100 MB de espacio para una instalación mínima de Linux. Para una instalación completa con todos los servicios, los requerimientos pueden ser tanto como 2 GB |
| Unidades de CD-ROM | Sistema estándar de archivo ISO 9660 para CD-ROMS |
| Unidades de Cinta | SCSI |
| Impresoras | Impresoras paralelas |
| Modems | Modems seriales internos y externos |
| Tarjetas Ethernet | Soporta tarjetas Ethernet y adaptadores LAN populares |

Distribuciones de Linux

Como el código fuente para Linux fue desarrollado esta siendo distribuido gratuitamente, diferentes compañías han desarrollado sus propias versiones o *distribuciones* de Linux

Cada una de estas variedades tiene su propio conjunto de características, tales como procedimientos de instalación y administración, paquetes de software y configuraciones. Muchas de ellas están configuradas para un tipo específico de computadora.

Las 10 distribuciones principales se listan a continuación:

- Mandrake Linux, desarrollado por MandrakeSoft.
- Red Hat Linux, desarrollado por Red Hat
- Debian GNU/Linux, desarrollado por Debian.
- SuSE Linux, desarrollado por SuSe, Inc.
- Gentoo Linux, desarrollado por Gentoo Technologies, Inc.
- El Proyecto Slackware Linux, desarrollado por Slackware Linux, Inc.
- Lycoris Desktop, desarrollado por Lycoris
- Beehive Linux, desarrollado por el Equipo Beehive
- Caldera OpenLinux, desarrollada por Caldera Internacional, Inc.
- Turbolinux, desarrollado por Turbolinux, Inc.

Existen muchas más.

5. BASH, EDICIÓN DE LÍNEA E HISTORIAL

* [Tab]: Autocompleta el nombre de un comando, archivo, directorio o programa, cuando trabajamos en una terminal texto. P.ej: Si vamos a usar el programa "programa_de_prueba", podemos escribir progr y pulsar [Tab], el sistema se encargara de rellenar el resto.

* [ArrowUp]: (Flecha arriba) Va editando la historia de comandos que hemos escrito anteriormente en terminal texto.[Enter] para ejecutar.

* [Shift][PgUp]: Scroll la salida de la terminal hacia arriba, en terminal texto.

* [Shift][PgDown]: Scroll la salida de la terminal hacia abajo, en terminal texto.

* [Ctrl]+c: Termina el proceso actual. Cuando lo comenzamos sin & ([localhost]\$ proceso)

* [Ctrl]+d: Termina la terminal actual.

* [Ctrl]+s: Para la transferencia a la terminal.

* [Ctrl]+z: Manda el proceso actual (comenzado sin &) a "Background". Lo mismo que si comenzamos el proceso con & ([localhost]\$ proceso &).

Lista de comandos más usados

- * hostname: Devuelve el nombre de la máquina.
- * uptime: Devuelve la cantidad de tiempo transcurrido desde la última vez que se arrancó el sistema, la cantidad de usuarios trabajando en el sistema y el load average (carga del sistema).
- * uname -a: Información sobre el sistema operativo de la máquina.
- * free -tm: Información sobre la cantidad de memoria disponible y usada.
- * df -h: Información sobre todo los dispositivos montados en la máquina.
- * du: Muestra el espacio que esta ocupado en disco.
- * du -bh /|more: Información sobre el espacio ocupado por cada subdirectorio, comenzando en el directorio raíz /.
- * ps: Información sobre los procesos del actual usuario, que se están ejecutando.
- * ps axu: Información sobre todos los procesos que se estan ejecutando en la máquina.
- * ping máquina: Para comprobar si tenemos contacto con la máquina máquina (máquina: nombre o IP)
- * /sbin/route: Tabla de enrutamiento de nuestro sistema.
- * /sbin/ifconfig: Información sobre los distintos dispositivos de red
- * netstat: Información valiosa sobre la conexión de red (Este comando tiene muchas posibilidades, ejecutar man netstat
- * ls: Lista archivos y directorios, funciona como el comando dir de DOS.
- * cp: Copia archivos/directorios.
- * rm: Borra archivos/directorios.
- * mkdir: Crea directorios.
- * rmdir: Borra directorios (deben estar vacíos).

- * mv: Renombrar o mover archivos/directorios.
- * date: Indica la fecha y hora del sistema.
- * pwd: Indica la ruta absoluta donde se está ubicado.
- * more: Muestra el contenido de un archivo con pausas cada 25 líneas.
- * grep: Filtra los contenidos de un archivo.
- * wc archivo: Muestra el número de palabras, líneas y caracteres de un archivo.
- * wc -c archivo: Muestra el tamaño en bytes de un archivo.
- * calendar: Recordatorio de fechas.
- * cal: Despliega un calendario.
- * cal -my: Calendario completo del año.
- * traceroute: Herramienta de red que nos muestra el camino que se necesita para llegar a otra máquina.
- * ifconfig: Configuración de interfaces de red, modems, etc.
- * su: Te conviertes en administrador(root) después de introducir la clave de acceso. Ideal para realizar alguna tarea de administración sin necesidad de salir del sistema y entrar de nuevo como root.
- * chmod permisos fichero/directorio: Cambia los permisos de ficheros/directorios
- * chown: Cambia los permisos de usuario: grupo de archivos/directorios.
- * sort: Ordena el contenido de un archivo.
- * tail: Muestra el final de un archivo.
- * head: Muestra la cabecera de un archivo.
- * file: Nos dice de que tipo es un archivo.
- * cp fichero1 fichero2: Copia fichero1 como fichero2

- * rm fichero: Borra fichero
- * rm -R directorio: Borra el contenido completo (ficheros/subdirectorios) de directorio
- * mv fichero1 fichero2: Cambia el nombre de fichero1 a fichero2
- * mkdir directorio: Crea un subdirectorio
- * rmdir directorio: Borra un subdirectorio
- * shutdown -t3 -r now: Reinicializa el sistema (hay que hacerlo como root).
- * shutdown -t3 -h now: Apaga el sistema (hay que hacerlo como root).
- * reboot: Reinicia el sistema.
- * poweroff: Apaga el sistema.

Historial de comandos

- history opción : Muestra los últimos comandos realizados, si se le pasa opción como un numero muestra los últimos solicitados.

6. GESTIÓN DE USUARIOS Y GRUPOS

- * /usr/sbin/adduser usuario: Registra y crea una cuenta de usuario.
- * /usr/sbin/userdel usuario: Borra la cuenta de usuario usuario.
- * passwd: Cambia la clave de acceso para el usuario actual. Root puede cambiar la clave de cualquier usuario passwd usuario
- * /usr/sbin/groupadd grupo: Crea un nuevo grupo.
- * /usr/sbin/groupdel grupo: Borra un grupo.
- * more /etc/passwd: Muestra el fichero de claves del sistema. (Si no se usa shadow password)
- * more /etc/group: Muestra los grupos registrados en el sistema.
- * finger usuario: Información sobre el usuario.

* who: Información sobre los usuarios usando el sistema.

* id usuario: Información sobre UID, GID y GROUPS del usuario.

* last: Información sobre los últimos usuarios que han usado el sistema.

7. MANEJO DE DIRECTORIOS Y FICHEROS

Creando directorios y Archivos

- mkdir <nombre_directorio>
 - mkdir dir1
 - mkdir dir1/dir2

Eliminando directorios y Archivos

- rm -[opciones] <nombre_archivo>
 - rm archivo1
 - rmdir dir1
- rm -[opciones] <nombre_archivo>

| <i>Opciones</i> | <i>Descripción</i> |
|-----------------|--|
| -r | Borra un directorio y todo su contenido recursivamente |
| -f | La opción -f es para que no pregunte al usuario por cada archivo que va a borrar ya que si se van a borrar 30 archivos resultaría muy tedioso colocar "yes" cada vez que pregunta si deseamos que descienda a través de los directorios y además pregunta si realmente deseamos borrar los archivos. |
| -i | Pregunta si realmente se desea borrar el archivo. |
| -v | A medida que elimina un archivo o directorio lo va indicando por pantalla. |

Moviendo y renombrando archivos y directorios

- `mv <archivo1> <archivo2>`
- `mv <archivo1> <directorio1>`
- `ln -s /etc mi_enlace`

8. SISTEMA DE PERMISOS

Todos los archivos y directorios en los sistemas Linux tienen una forma estándar de acceso. Estos permisos de accesos controlan cuáles archivos pueden ser acceder por quién. Para ver los permisos de los archivos y directorios se puede usar el comando `ls -l`.

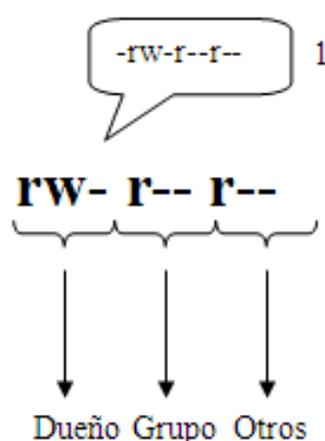
```
$ ls -l /home/cursoscc/prueba
-rw-r--r-- 1 usuario1  grupo1  1319  Mar 15 11:23 prueba
```

rW- **r**-- **r**--

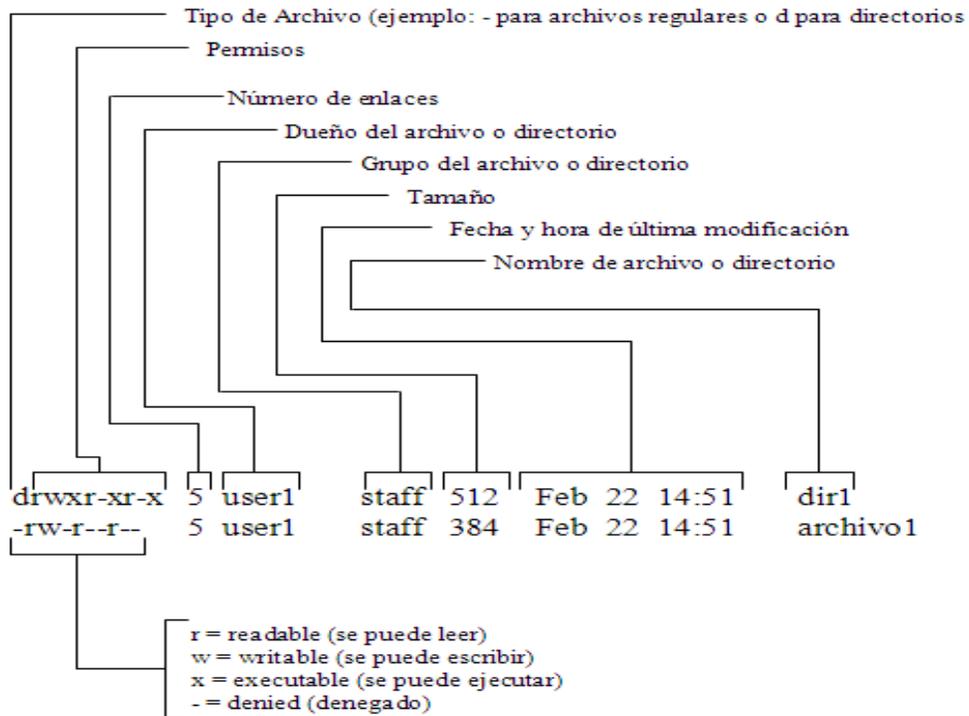
┌───┬───┬───┐

↓ ↓ ↓

Dueño Grupo Otros



Visualización de permisos sobre un archivo



Información detallada de archivos

Tipos de Usuario

| <i>Campo</i> | <i>Descripción</i> |
|--------------|---|
| Dueño | Permisos usados por el dueño del archivo o directorio. |
| Grupo | Permisos usados por los miembros del grupo que posee el archivo o directorio. |
| Otros | Permisos usados por todos aquellos usuarios que no son dueños ni miembros del grupo al que pertenece el directorio o archivo. |

| <i>Permisos</i> | <i>Carácter</i> | <i>Acceso para archivos</i> | <i>Acceso para directorios</i> |
|-----------------|-----------------|--|--|
| Lectura | r | Puede ver el contenido del archivo y puede copiarlo. | Puede listar el contenido del directorio con el comando ls. |
| Escritura | w | Puede modificar el contenido del archivo. | Si posee también permisos de ejecución, puede modificar el contenido del directorio. |
| Ejecución | x | Puede ejecutar el archivo si éste fuera ejecutable. Puede ejecutar un shell script si tuviera también permisos de lectura y escritura. | Puede usar el comando cd para acceder al directorio. Si también tiene permiso de lectura puede ejecutar ls -l para listar el contenido del directorio. |

OPCIONES CON EL COMANDO: ls

| <i>Opción</i> | <i>Descripción</i> |
|---------------|---|
| -l | Hace un listado en formato largo, dando información de sus permisos, propietario, etc. |
| -a | Muestra los archivos ocultos (los que empiezan por .) |
| -F | Identifica los tipos de archivos, coloca al final del nombre uno de los siguientes símbolos: / si es un directorio * si es un ejecutable @ si es un enlace simbólico |
| -t | Lista los archivos y directorios por fecha de modificación. |
| -h | Muestra el tamaño de los archivos en KB o MB. |
| -n | Lista los UID y GID del usuario en vez de los nombres. |
| -R | Lista el contenido de un directorio recursivamente. |
| -d | Lista los permisos del directorio que se pase por parámetro. |

Con los ocho dígitos, ocho combinaciones diferentes son posibles para cada categoría.

0 - - - ni lectura, ni escritura, ni ejecución (permiso sin sentido)

1 - - x solo ejecución

2 -w -- solo escritura

3 – wr escritura y ejecución

4 r - - solo lectura

5 r – x lectura y ejecución

6 rw- lectura y escritura

7 rwx lectura, escritura y ejecución

r indica lectura (read), w indica escritura (write) y x indica ejecución (execute).

Se presenta un ejemplo. El administrador del sistema quiere dar el siguiente permiso:

- Todos los permisos al usuario
- Permiso de lectura y ejecución para el grupo al que pertenece el usuario
- Permiso de solo lectura para los otros

Los permisos en octal se escribirán como sigue:

- 7 para el usuario
- 5 para el grupo de usuarios
- 4 para los otros.

De este modo el permiso se escribe 754

Linux proporciona el comando **chmod** (siglas en inglés de change mode) para asignar permisos a los archivos y directorios, además de emplearse para cambiar los permisos ya otorgados.

Sintaxis de chmod: chmod [opcion] permiso archivo
uneweb@localhost :~> chmod 754 miarchivo.tx

El comando anterior otorga permisos de lectura, escritura y ejecución al Usuario; lectura y ejecución al Grupo de Usuarios y solo lectura para los otros.

Pero no tiene sentido dar permisos de ejecución a un archivo de texto. Los archivos de texto solo pueden ser leídos y modificados. Ejecutar un archivo es correr un programa o comando.

9. EDICIÓN CON VI

Vi es un poderoso editor de texto de la línea de comandos, es uno de los más viejos que se ha usado durante generaciones en Linux.

Este editor tiene tres modos básicos de operación:

- Modo comando
- Modo de edición
- Modo de última línea

| <i>Opciones</i> | <i>Descripción</i> |
|-----------------|---|
| i | Insertar texto a la izquierda del cursor. |
| a | Insertar texto a la derecha del cursor. |
| I | Insertar al comienzo de la línea. |
| A | Insertar al final de la línea. |
| o | Añade una línea después de donde se encuentra el cursor. |
| O | Añade una línea antes del cursor. |

| <i>Opciones</i> | <i>Descripción</i> |
|-----------------|--|
| u | Deshacer la última acción |
| U | Deshacer los últimos cambios que se hicieron en la línea. |
| dd | Borra la línea completa donde se encuentra el cursor. |
| Ndd | Borra N líneas a partir de donde está el curso (N es un número dado por usted). |
| yy | Copia la línea donde está el cursor y las coloca en el buffer. |
| P | Pega lo que está en el buffer en la línea siguiente donde está el cursor. |

| | |
|------------|---|
| P | Pega lo que está en el buffer en la línea anterior donde está el cursor. |
| dw | Borra desde donde está el cursor hasta el final de la palabra. |
| d\$ | Borra desde donde está el cursor hasta el final de la línea. |
| NG | Un número de línea seguido de G mueve el cursor a la línea con ese número. |
| . | Rehace la última modificación. |

| <i>Opciones del Modo última línea</i> | |
|---------------------------------------|---|
| :x | Guarda y sale del editor vi. |
| :q | Sale si no hubo cambios. |
| :q! | Sale sin guardar los cambios. |
| :w | Guarda los cambios. |
| :w <nombre_archivo> | Guarda el archivo con el nombre <nombre_archivo>. |
| :# Línea | Nos lleva el número de línea especificado por #. |
| :set un | Enumera las líneas. |
| :set all | Muestra todas las variables de vi y sus respectivos valores asociados. |
| :r <archivo> | Carga el contenido de otro archivo y lo inserta en el archivo actual. |

10. FILTROS Y REDIRECCIONES

La redirección en Linux ayuda a conectar programas con archivos, mientras que las tuberías ayudan a conectar programas con otros programas. Las tuberías en línea es una característica donde la salida de un programa es enviada como entrada para otro programa. El carácter | (una barra vertical) representa una tubería.

Los caracteres `>`, `<` y `>>`. Estos se denominan metacaracteres de redirección y `|` metacaracter de tubería. A continuación dos usos adicionales de redirección, `n>` `>&n`.

Para mostrar los usos se asume que el archivo `linux.txt` contiene las siguientes líneas.

```
Linux es un sistema operativo
Sigue los estándares GNU.
Linux esta provisto de un lenguaje de comandos
El kernel de Linux es el núcleo del sistema operativo
Linux es un poderoso y versátil sistema operativo.
```

```
uneweb@localhost:~> cat linux.txt; date > datefile
```

Entonces, solo la salida `date` se habrá almacenado en `datefile`. Agrupar comandos ayuda a combinar la salida de más de un comando en un archivo de salida.

```
uneweb@localhost:~> cat lineas.txt | grep esper | wc -l
3
uneweb@localhost:~>
```

La salida `cat` es enviada como entrada a `grep`. El comando `cat` no procesa datos. Simplemente envía la entrada a la salida estándar. Pero antes que pueda mostrarse en la salida estándar, se encuentra el símbolo tubería (`|`), que ayuda al shell a entender que la salida debe ser enviada como entrada al siguiente comando en la tubería en línea.

El comando `grep` toma la entrada (y por lo tanto no se da como un archivo como argumento) y lo procesa buscando el patrón `esper` en cada línea, además envía la salida.

Esta salida es pasada nuevamente a través de la tubería a `wc`, que cuenta el número de líneas y finalmente lo que se muestra es 3.

El uso de tubería presentado encuentra que existen tres líneas en el archivo `lineas.txt` que tienen el patrón `esper` en ellas.

```
uneweb@localhost:~> head -5 lineas.txt | grep esper | sort
```

No se muestra salida alguna, dado que el patrón `esper` no existe en las cinco primeras líneas del archivo `lineas.txt`.

Un filtro es un programa que lee datos de la entrada estándar, los procesa y envía los datos procesados a la salida estándar

El comando `sort` es un ejemplo de un filtro. Toma la entrada de la entrada estándar (ya sea a través del teclado o un archivo), procesa los datos (los ordena en el orden que se requiere) y los envía a la salida estándar (ya sea la pantalla o un archivo, si se usa redirección).

Sin embargo, no todos los comandos en Linux son filtros. Por lo tanto, es importante notar que el lado derecho de una tubería puede no siempre ser un filtro. Por ejemplo, `cal no` es un filtro y su uso en `cat nombres.txt | cal no` es un uso válido en tuberías y filtros.

En situaciones como estas, el último comando se ejecuta y todos los demás datos se pierden en la tubería. Cuando estos son incorrectos se hacen con tuberías y filtros en un programa, el error mismo puede ser difícil de identificar y corregir. Por lo tanto, se debe ser cuidadoso para asegurar que el uso en el lado derecho de una tubería sea un filtro correcto.

Comando grep

Asuma que este archivo tiene las siguientes líneas:

Los filtros son programas que leen la entrada, la procesan y la escriben en la salida. Algunos ejemplos de filtros son `sort`, `tail`, `head` y `grep`. Aquí tenemos más filtros en Linux. `grep` es un poderoso filtro que busca patrones específicos y los imprime en pantalla.

```
/home/uneweb$ grep filtros filtros.txt
```

El uso anterior de `grep` es el más sencillo. Encuentra el patrón `filtros` en el archivo `filtros.txt` y muestra las líneas que coinciden con este.

```
/home/uneweb$ grep ^filtros filtros.txt
```

Muestra solo las líneas que tienen el patrón `filtro` al inicio de la línea.

```
/home/uneweb$ grep filtros$ filtros.txt
```

Muestra solo las líneas que tienen el patrón `filtro` al final de la línea.

```
/home/uneweb$ ls -l | grep '^d'
```

Imprime los nombres de los subdirectorios del directorio de trabajo actual. Se obtienen los nombres de los subdirectorios como resultado del comando anterior, esto se debe a que el listado completo muestra los permisos de los archivos y si el archivo es un directorio, la línea comienza con el carácter `d`. El comando anterior imprimirá las líneas que contienen el carácter `'d'`. Las comillas simples se usan para quitar el significado especial de `^`.

También se puede usar `[list]` y `^[list]` en `grep`. Por ejemplo considere lo siguiente:

```
/home/uneweb$ ls -l | grep [5-6]
```

Equivale a todas las líneas que tienen `5` o `6` en ellas. Las demás son ignoradas.

```
/home/uneweb$ ls -l | grep f
```

Esto equivale a todas las líneas que tienen por lo menos una ‘f’ en ellas e ignora aquellas que no. Se obtiene el resultado como se declaro, siempre que exista al menos un archivo en el directorio que tiene f en su nombre. En caso contrario no muestra nada.

| Carácter Especial | Significado |
|---------------------------|---|
| . | Equivale a cualquier carácter único. r. n coincidirá con run, ran, rbn, ron etc.. Nota: el . (punto) significa un solo carácter entre r y n. La línea puede tener caracteres antes de r y después de n. Así el patrón relaciona abrun, ranbb o abrnakk |
| [] (corchetes) | Equivale a cualquier carácter especificado en el rango. [1234] book coincidirá con 1book, 2book, 3book, 4book |
| [^] | Equivale a cualquier carácter que no este en el rango [^abc] equivale a todas las líneas que contienen cualquier carácter que no este en [a, b, c]. Así adf y kkkkc coincidirán, pero no aabc o aaaa |
| ? | Equivale a cero o una ocurrencia del carácter precedente, ab? Coincidirá con a y b |
| * (asterisco) | Equivale a cero o mas ocurrencias del carácter precedente, ab* equivale a, ab, abb, abbbb, aaa, etc, aaa y acc también coinciden porque el * relaciona cero o mas ocurrencias del carácter precedente, que es b en este caso. |
| + | Equivale a una o más ocurrencias del carácter precedente, ab+ coincide con ab, abb, abbbb, etc,. No coincide con aaa y acc ya que + indica una o mas ocurrencias del carácter precedente de b. |
| ^ (acento circunflejo) | Equivale al inicio de la línea. Por ejemplo, ^this coincide con todas las líneas que tienen this como patrón de inicio de líneas. |
| \$ | Equivale al final de la línea. Por ejemplo, \$this coincide con todas las líneas que tienen this como patrón de finalización de líneas. |

11. FIND

Comando para buscar archivos y directorios

```
find ruta expresión acción
find /home -name <nombre_archivo>
find / -user <nombre_usuario>
```

12. CONTROL DE PROCESOS

[Un proceso se define como ‘una instancia de un programa en ejecución’

Comando ps.

Se puede conocer el estado de un proceso usando el comando ps, donde ps equivale a process status (estado del proceso)

```
/home/uneweb$ ps
PID TTY TIME CMD
734 pst/1 00:00:12 bash
754 pts/1 00:00:01 ps
```

PID muestra el número de identificación del proceso. Cada proceso obtiene un número de identificación único asignado por el kernel

TTY es el tipo de terminal asociado con el proceso. También puede ser otro dispositivo de entrada conectado al sistema.

TIME es el tiempo que el proceso ha estado en ejecución hasta ahora

CMD es el nombre del comando que está siendo ejecutado, cuyo reporte de **estado** se está visualizando. Bash es el proceso shell.

```
/home/uneweb$ ps -A
```

La opción `-A` muestra los otros procesos ejecutándose en el sistema. El comando básico `ps` solo muestra la ejecución de los procesos del usuario actual.

```
/home/uneweb$ ps r
```

La opción `-r` muestra los detalles para el proceso ejecutándose actualmente.

También se puede ver un listado largo del comando `ps`, usando la opción `-l`. El comando `ps -l` proporciona el reporte de estado.

Asumiendo que lo siguiente es la salida del comando `ps`, se intentará matar uno de los procesos en ejecución.

```
/home/uneweb$ ps
PID TTY TIME CMD
970 pts/3 00:00:00 bash
1072 pts/3 00:00:00 bash
1103 pts/3 00:00:00 ps
```

```
/home/uneweb$ kill 1072
```

Si se ingresa `ps` nuevamente, probablemente se verá una salida similar a la anterior, ya que el comando `kill` no ha terminado el segundo proceso `bash`.

```
/home/uneweb$ ps
PID TTY TIME CMD
970 pts/3 00:00:00 bash
1072 pts/3 00:00:00 bash
1103 pts/3 00:00:00 ps
```

```
/home/uneweb$ kill -9 1072
killed
```

```
/home/uneweb$ ps
PID TTY TIME CMD
970 pts/3 00:00:00 bash
1103 pts/3 00:00:00 ps
```

Ahora se observa que el proceso bash hijo ha muerto. Esta es otra forma de salir del programa del proceso bash hijo

13. INTRODUCCIÓN A LA PROGRAMACIÓN SHELL

El shell interpreta los comandos ingresados por los usuarios y los comunica al núcleo (kernel). El kernel maneja las interacciones complejas con el hardware del sistema.

El shell actúa como un intermediario entre el usuario y el kernel.

Algunas de las responsabilidades del shell son:

- Controlar y asignar trabajos
- Manejar mas de un proceso a la vez
- Redireccionar entrada y salida estándar
- Mantener un historial de comandos
- Proporcionar un lenguaje de comandos para escribir shell scripts

Todos los scripts Bash tienen que comenzar con la línea: `#!/bin/bash`

Esta línea le indica al shell usado por el usuario, el que tengamos activo en el momento, con qué shell hay que ejecutar el script que aparezca a continuación.

El script puede ejecutarse de dos modos diferentes:

- 1) Por ejecución directa desde la línea de comandos, siempre que tenga permiso de ejecución. Si no se da el caso, ponemos el permiso con: `chmod +x script`.
- 2) Por ejecución mediante el shell, llamamos al shell explícitamente: `bash script`

Variables Shell

Cada shell tiene su propio conjunto de variables

Se puede ver el contenido de todas las variables shell ingresando un símbolo \$ (de peso), antes de la variable. De esta manera

```
uneweb@localhost:~> echo $PATH
/opt/java/bin:/opt/java/bin:/home/uneweb/bin:/usr/local/bin:/usr/X11R6/bin:/usr/gams/opt/
kde/bin:/gnome/bin:/usr/lib/mit/sbin:/usr/lib/qt3/bin:/opt/IBMJava2142/bin:/usr/local/lib/xerces-c-src_2_7_0/bin:/usr/local/pgsql/bin
```

Lo anterior es el contenido de la variable PATH. El contenido podría ser diferente en otros sistemas. Cada ruta de directorio está separada por dos (:) puntos.

PATH es solo una de muchas variables predefinidas que mantiene un shell. En la Unidad

Variables Shell Definidas por el Usuario

Para crear una variable, el usuario debe ingresar:

```
name=value
```

Note que no debe existir espacio antes ni después del signo de igualdad (=).

```
uneweb@localhost:~> color=azul
```

```
uneweb@localhost:~>.
```

La variable color ahora es definida por el usuario y le es asignado el valor azul.

Para ver el valor de la variable shell definida por el usuario, se usa el signo \$ junto con el nombre de la variable. Usando el comando echo, se puede imprimir el valor establecido por el usuario.

```
uneweb @localhost:~> echo $color
```

```
azul
```

Los nombres de las variables definidas por el usuario no pueden comenzar con un número

Sustitución de Parámetros

El shell proporciona un método para definir y sustituir variables usando metacaracteres de sustitución de parámetros { }. Las siguientes son cuatro formas en las que se puede hacer esto:

`${variable-word}` Muestra el valor de variable si este existe; en caso contrario muestra el valor word

`${variable=word}` Muestra el valor de la variable si este existe; en caso contrario muestra el valor de word y asigna a variable el valor de word

`${variable+word}` Si variable esta asignada, muestra word, en caso contrario no muestra nada

`${variable?mesg}` Si variable ya esta asignada, muestra el valor, en caso contrario muestra mesg.

Uso de Sustitución de Parámetros

```
uneweb @localhost:~> echo $ {color-verde}
```

```
verde
```

Como color no esta definido, se muestra verde

```
uneweb @localhost:~> color=azul
```

```
uneweb @localhost:~> echo $ {color-naranja}
```

```
azul
```

Muestra azul dado que color ya esta definido. En ambos casos la variable color permanece sin cambios

```
uneweb @localhost:~> echo $ {nuevolor-verde}
verde
uneweb @localhost:~>
```

Muestra verde y le asigna a nuevolor el valor verde

```
uneweb @localhost:~> echo $ {nuevolor+amarillo}
amarillo
```

Como nuevolor ya esta asignado, se muestra amarillo, nuevolor no es modificado

```
uneweb @localhost:~> echo $ {otro+amarillo}
uneweb @localhost:~>
Como otro no esta asignado no muestra nada
```

```
uneweb @localhost:~> echo $ {nuevolor? no definido}
verde
```

Como la variable color tiene valor asignado, muestra el valor de nuevolor

```
uneweb @localhost:~> edwar] $ echo $ {otro? no definido}
bash: otro no definido
```

El mensaje no definido es porque la variable otro no esta definida hasta ahora

14. ARRANQUE DEL SISTEMA.

La BIOS busca un dispositivo de inicio (disco duro, CDRROM,..) y pasa el control al MBR (512 bytes).

- Se carga el gestor de arranque (instalado en MBR).
- Se carga el kernel.
- Se monta el sistema de ficheros raiz (/).
- Se inicia el init (el abuelo de todos los procesos).
- Se lee el archivo /etc/inittab.
- Se ejecutan los scripts indicados por el nivel de ejecución de arranque.

Gestores de arranque

Grub

- Grub (Grand Unified Bootloader) es un gestor de arranque capaz de arrancar diferentes sistemas operativos en diferentes particiones y discos duros
- Normalmente se instala en el MBR (master boot record) del disco duro principal.
- El archivo de configuración en Debian GNU/Linux es:
/boot/grub/menu.lst
- No es necesario ejecutar ningún comando para que se tengan en cuenta los cambios realizados.

15. NIVELES DE EJECUCIÓN

En los sistemas Unix hay 7 (0...6) niveles de ejecución

- nivel 0: estado de parada (halt)
- nivel 1: monousuario
- nivel 2, 3 y 5: multiusuario
- nivel 4: no tiene un uso específico
- nivel 6: estado de reinicio (reboot)

Init

- Es el primer proceso que se crea (PID=1).
- Se configura mediante /etc/inittab.

→ Sintaxis

id:nivel:accion:proceso

- id: nombre de la línea
- nivel: nivel o niveles en los que la línea debe procesarse
- acción wait (espera), once (solo 1 vez), respawn (rearranca una vez finalizado), off (ignora la línea)
- proceso: path del proceso a ejecutar

→ Valores típicos del archivo /etc/inittab:

```
id:2:initdefault
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
z6:6:respawn:/sbin/sulogin
# CTRLALTDDEL!
ca:12345:ctrlaltdel:/sbin/shutdown t1 a r now
```

Cuando la máquina se inicia en un nivel de ejecución N se ejecutan todos los scripts que empiecen por 'S' o 'K' contenidos en /etc/rcN.d/ con las siguientes particularidades.

→ Si el nombre del script comienza por S se le pasa como parámetro 'start'.

→ Si el nombre del script comienza por K se le pasa como parámetro 'stop'.

- Normalmente todos estos ficheros de /etc/rcN.d son enlaces a scripts localizados en /etc/init.d/

16. CONFIGURACIÓN DE RED

Las interfaces de red se configuran en el fichero '/etc/network/interfaces' (man interfaces).

```
aktor@irontec:~$ cat /etc/network/interfaces
auto eth0 eth1
iface eth0 inet static
address 192.168.0.2
netmask 255.255.255.0
broadcast 192.168.0.255
gateway 192.168.0.1
iface eth1 inet dhcp
```

Configuración manual de la red

- ifconfig: configura interfaces de red. Asigna IP, máscara, gateway, etc.

```
$ ifconfig eth0 192.168.0.12 netmask 255.255.255.0 up
```

- route: añade rutas estáticas. Ejemplo típico:

```
$ route add default gw 192.168.0.1
```

```
$ route add net
```

```
10.10.0.0 netmask 255.255.255.0 gw
```

```
192.168.0.100
```

```
$ route n (muestra tabla de rutas)
```

- /etc/resolv.conf: indica los servidores DNS

```
$ cat /etc/resolv.conf
```

```
nameserver 195.235.113.3
```

- /etc/hosts: realiza una resolución directa de nombre a IP sin realizar consulta DNS. Normalmente este fichero se mira antes de preguntar al servidor DNS (sólo para consulta de tipo A).

- Para asegurarnos que el sistema pregunte al DNS o al fichero '/etc/hosts' para resolver nombres de host hay que mirar el fichero '/etc/nsswitch.conf'

Configuración automática de la red (DHCP)

- Configuración para el arranque: en /etc/network/interfaces definimos la interfaz como "inet dhcp"

- Configuración manual:

→ se utiliza un cliente dhcp como dhclient o pump

```
# dhclient eth0
```

```
# pump i
```

```
eth0
```

17. ACCESO REMOTO AL SISTEMA SSH

INSTALAR-Y-CONFIGURAR-SSH-EN-DEBIAN/

-- Actualizamos e instalamos SSH:

```
# apt-get update
```

```
# apt-get install ssh
```

--- Configuramos la seguridad de SSH:

-Editamos el fichero de configuración:

```
# vi /etc/ssh/sshd_config
```

-Parámetro para permitir logins a root, de diremos que no, ya que nos conectaremos con otro usuario, y utilizaremos SUDO para tareas administrativas, Modificamos el "yes" por un "no"

```
PermitRootLogin no
```

-Tendríamos que modificar el puerto por defecto, poniendo por ejemplo:

```
Port 2222
```

-El tipo de protocolo usado, por defecto debería de ser la versión 2:

```
Protocol 2
```

-El tiempo que esperara en el login si no le damos un usuario, es un timeout (debería de ser bastante bajo):

```
LoginGraceTime 15
```

-Máximos intentos erróneos al iniciar la sesión de login:

```
MaxAuthTries 2
```

-Añadimos al final del fichero que usuario permitiremos acceder por SSH:

```
AllowUsers usuario1 usuario2 usuario3
```

---Reiniciamos servicio:

```
# /etc/init.d/ssh restart
```

Con todo esto ya tenemos instalado y configurado nuestro servicio SSH seguro, ahora procedemos a realizar la conexión desde un entorno Linux y un entorno Windows con putty.

--- Probamos desde otro Linux:

```
# ssh -p puerto usuario@ip_maquina
```

--- Probamos desde un Windows (putty)

SCP

- Comando para copiar ficheros sobre SSH:

→ Copiar “fichero” en la home de “usuario” en servidor:

```
scp fichero usuario@servidor:
```

→ Copiar un directorio recursivamente en /tmp del servidor:

```
scp -r dir usuario@servidor:/tmp
```

→ Copiar fichero2 desde el servidor a mi home:

```
scp usuario@servidor:fichero2 ~
```

→ Copiar de máquina B a máquina C:

```
scp usuarioB@servidorB:fichero usuarioC@servidorC:
```

Ejemplo 1

```
scp imagen001.jpg 192.168.1.105:imagen001-copy.jpg
```

Copia el archivo imagen001.jpg del directorio actual, al directorio home del mismo usuario, en la máquina 192.168.1.105 bajo el nombre imagen001-copy.jpg

Ejemplo 2

```
scp 192.168.1.105:imagen001-copy.jpg ./imagen001-copy2.jpg
```

Copia el archivo imagen001-copy.jpg del directorio home en la máquina 192.168.1.105, del usuario logueado, al directorio actual, bajo el nombre imagen001-copy2.jpg

Ejemplo 3

```
scp archivo1.c marco1@192.168.122.130:/home/marco1/programas/
```

Copia el archivo archivo1.c del directorio actual, al directorio /home/marco1/programas/, utilizando el usuario marco1.

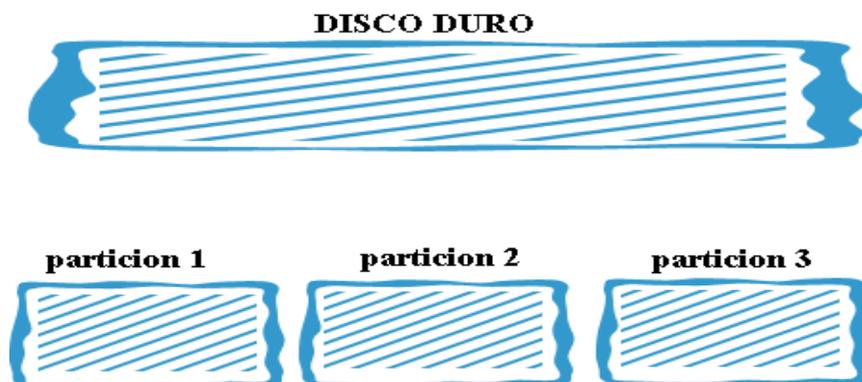
Ejemplo 4

```
scp -r marco1@192.168.122.130:/home/marco1/codigo/marco2@192.168.122.131:/home/marco2/codigo/
```

Copia recursivamente el contenido del directorio /home/marco1/codigo del usuario marco1, en la máquina 192.168.122.130 hacia el directorio /home/marco2/codigo/ del usuario marco2, en la máquina 192.168.122.131.

18. CREACIÓN Y USO DE SISTEMAS DE ARCHIVOS

- Las particiones, por decirlo de alguna manera son divisiones de los discos duros, es como si a un disco duro le hiciéramos varios cortes, y cada uno de los trozos que salieran se comporten como discos diferentes.
- Aunque seguimos teniendo un solo disco duro cada una de estas particiones funcionara de manera independiente.



Particionar un disco es como dividirlo en discos más pequeños contenidos dentro de la misma carcasa.

Las particiones pueden ser de dos tipos: primarias ó lógicas, las particiones lógicas se crean dentro de una partición primaria especial llamada extendida.

Como mínimo en un disco debe de haber una partición primaria y como máximo puede haber cuatro particiones primarias ó tres primarias y una extendida, no puede haber más de una partición extendida en un mismo disco duro.

Dentro de una partición extendida puede haber todas las particiones lógicas que queramos.

Cada partición puede contener un sistema de archivos diferentes.

Un sistema de archivos establece los mecanismos para acceder a los datos que contiene y permite o no el acceso a dichos datos en determinadas circunstancias y en caso de alguna inconsistencia posibilita medios para su restauración.

Existen diferentes sistemas de archivos y cada sistema operativo soporta diferentes sistemas de ficheros, algunos de ellos son los siguientes:

- **Ext2** el primer sistema de archivos diseñado expresamente para Linux.
- **Ext3** versión mejorada de Ext2, ampliamente utilizada en distribuciones Linux

- **Ext4** nuevo sistema de ficheros para Linux evolucionado de Ext3 algunas de sus ventajas con respecto a este último son:
 - Aumento del tamaño del sistema de ficheros a 1 Exabay = 2^{20} Terabyte y del tamaño máximo por archivo (16 Tb).
 - Número ilimitado de directorios frente a los 32.000 de Ext3
 - Mayor velocidad en la creación y borrado de ficheros
- **XFS** Implementado para sistemas Unix, rápido acceso para grandes cantidades de datos y altas velocidades, pero lento en ficheros pequeños. Se utiliza en grandes servidores y donde se maneje mucha información.
- **ReiserFS** Sistema de archivos diseñado e implementado por la empresa Namesys, liderado por Hans Reiser. Es soportado por Linux y otros sistemas operativos (algunos a través de implementaciones muy inestables).
- **Fat16/Fat32** Sistemas Dos/Windows 95, Windows 98, este sistema de archivos produce una gran fragmentación del disco.
- **NTFS** sistema de archivos implementado para los sistemas operativos Windows Xp/2000/2003 server y Vista.

A la acción de crear un sistema de ficheros dentro de una partición se le llama también formatear la partición.

La nomenclatura para identificar las particiones es diferente en Windows y en Linux. En Windows las unidades se identifican con letras, por ejemplo:

Primer disco duro (IDE, SCSI ó SATA)

C: (primera partición del primer disco duro IDE)

D: (segunda partición del primer disco duro IDE)

Segundo disco duro (IDE ó SCSI ó SATA)

E: (primera partición del segundo disco duro IDE)

F: (segunda partición del segundo disco duro IDE)

En Linux se representan de la siguiente manera:

hda (primer disco duro IDE).

hda1 (primera partición del primer disco duro IDE)

hda2 (segunda partición del primer disco duro IDE).

hda5 (primera partición lógica de una partición extendida del primer disco duro IDE)

hdb (segundo disco duro IDE).

hdb1 (primera partición del segundo disco duro IDE).

hdb2 (segunda partición del segundo disco duro IDE).

hdb5 (primera partición lógica de una partición extendida del segundo disco duro IDE)

Discos SCSI ó SATA.

sda (primer disco duro SCSI)

sda1 (primera partición del primer disco SCSI).

sda2 (segunda partición del primer disco SCSI).

sda5 (primera partición lógica del primer disco duro SCSI)

sdb (segundo disco duro SCSI)

GParted es un editor de particiones para Linux con el que podemos crear, borrar y redimensionar particiones. También podemos crear el sistema de ficheros que tendrá cada partición. Con la versión actual de Gparted 0.4.3 podemos crear los siguientes sistemas de ficheros:



ext2
ext3
ext4
fat16
fat32
hfs
hfs+
jfs
linux-swap
ntfs
reiser4
reiserfs
ufs
xfs
sin formatear

GParted Es muy fácil e intuitivo de usar, muchas distribuciones Linux lo incluyen en su live CD, entre ellas Ubuntu.

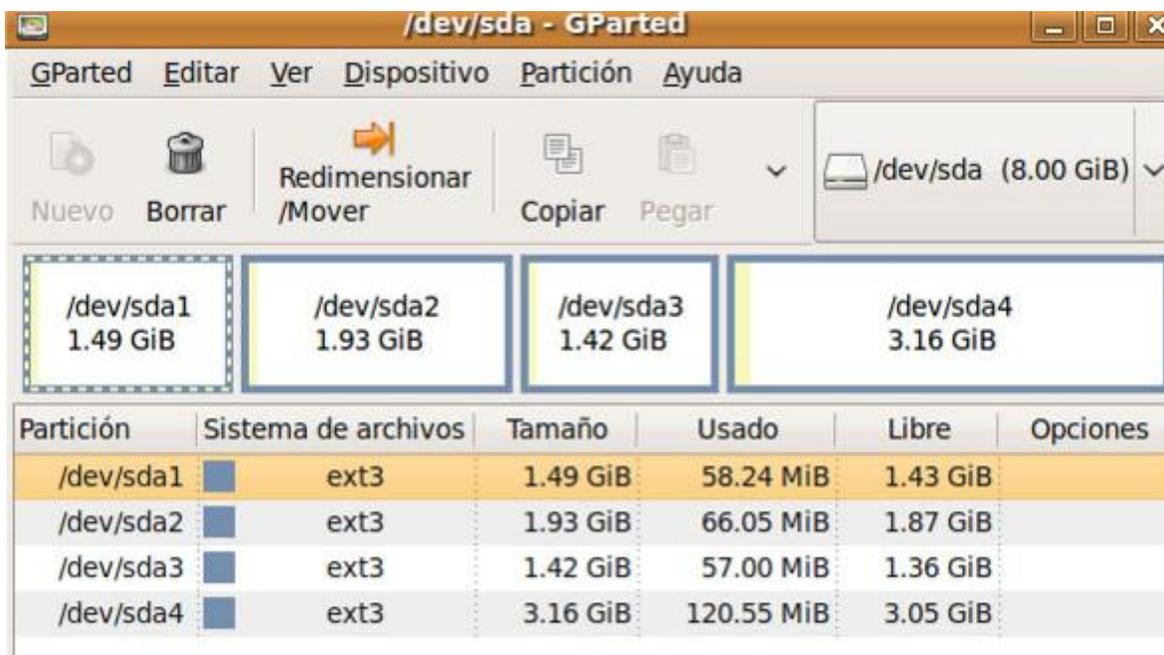
Una vez que tenemos creadas nuestras particiones en nuestro disco y formateadas con el sistema de archivos que van a utilizar, para poder utilizar dichas particiones en Linux, tenemos que hacer un paso más, que es crear como mínimo el punto de montaje para el directorio raíz (/). Linux utiliza una estructura de directorios jerarquizada en forma de árbol invertido en donde todo es tratado como si fuera un archivo y para poder acceder a él debe estar acoplado (montado) dentro de esa estructura de directorios encabezada por el directorio raíz.

La interfaz gráfica de instalación de cualquier “distro” de linux nos permite montar el directorio raíz del sistema donde se instalarán los archivos más importantes del sistema operativo, pero también podemos crear otros puntos de montaje como /home (que es como la carpeta Mis documentos de Windows pero mucho más potente) en otra partición diferente, de esta manera si formateamos la partición donde está el sistema operativo, mis documentos (tú /home) quedará a salvo y tus datos no se perderán.

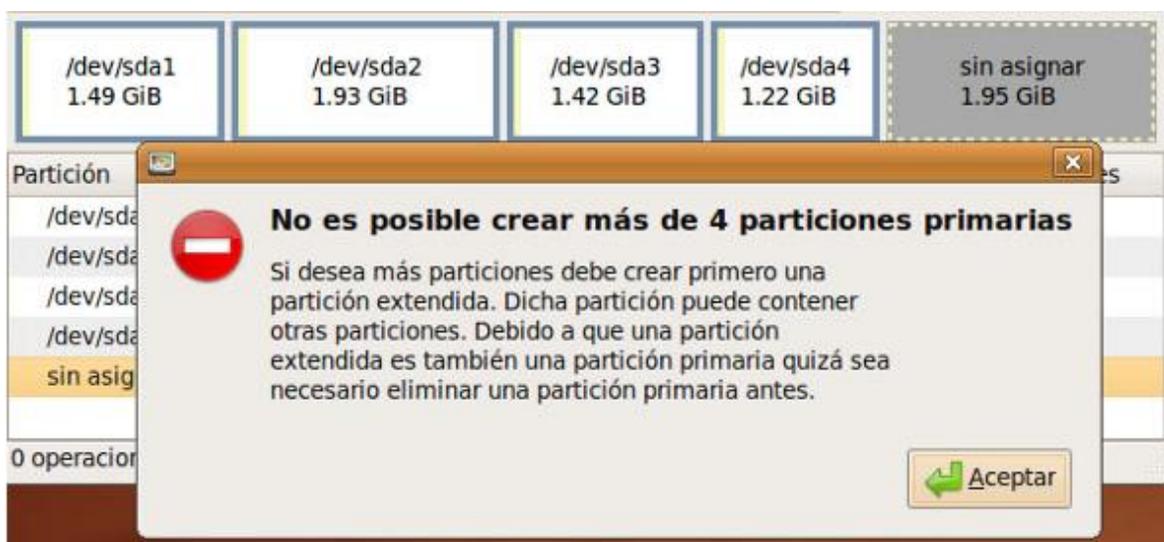
En Linux todo es tratado como un archivo, es decir, si tenemos una disquetera, un lector de DVD, un disco externo, etc. para que podamos acceder a estos dispositivos antes tienen que estar montados como un archivo más.

Ejemplos:

En la figura de abajo se muestra por medio del programa GParted un disco SCSI de 8 Gb que está dividido en cuatro particiones primarias formateadas con el sistema de archivos ext3 cada una de ellas.



En la siguiente figura vemos lo que pasa cuando intentamos crear más de cuatro particiones primarias.



En la siguiente figura vemos el mismo disco con tres particiones primarias (sda1, sda2 y sda3), una extendida (sda4) que contiene a su vez tres particiones lógicas: sda5 (formateada con el sistema de archivos fat32), sda6 (formateada con ntfs) y sd7 (utilizada para la memoria de intercambio swap de Linux).

| Partición | Sistema de archivos | Tamaño | Usado | Libre | Opciones |
|-------------|---------------------|------------|-----------|------------|----------|
| /dev/sda1 | ext3 | 1.49 GiB | 58.24 MiB | 1.43 GiB | |
| /dev/sda2 | ext3 | 1.93 GiB | 66.05 MiB | 1.87 GiB | |
| /dev/sda3 | ext3 | 1.42 GiB | 57.00 MiB | 1.36 GiB | |
| ▽ /dev/sda4 | extended | 3.16 GiB | --- | --- | |
| /dev/sda5 | fat32 | 949.12 MiB | 1.87 MiB | 947.25 MiB | |
| /dev/sda6 | ntfs | 1.13 GiB | 6.28 MiB | 1.13 GiB | |
| /dev/sda7 | linux-swap | 1.10 GiB | --- | --- | |

0 operaciones pendientes

Para la instalación de Linux, ya sea de la distribución que sea, se necesita como mínimo dos particiones: la principal, donde se instalará el sistema operativo, formateada con el sistema de archivos que permita la distribución que vayamos a instalar y otra partición llamada swap (intercambio de memoria virtual), esta partición es utilizada como memoria RAM virtual cuando tenemos muchas aplicaciones abiertas y la memoria RAM de nuestro ordenador es insuficiente, de tamaño del swap se suele poner el doble de la memoria RAM, es decir, si tenemos 250 Mb de memoria RAM el tamaño del swap debería de ser aproximadamente de 500 MB, pero estas cifras están pensadas para equipos con poca memoria. Por ejemplo, si te has comprado un equipo nuevo con 4GB de memoria RAM, no es obligatorio que ponga 8GB para el swap. También hay que considerar que si instalas Linux en un equipo con poca memoria y haces mucho uso del swap, tu ordenador se volverá más lento ya que se tarda más tiempo en leer y escribir del disco duro que de la memoria física RAM, en este caso deberías reconsiderar el ampliar la memoria RAM de tu ordenador.

19. INSTALACIÓN Y ACTUALIZACIÓN DE SOFTWARE.

aptget: interfaz para instalar y desinstalar paquetes así como para la generación y actualización de la base de datos local de paquetes.

→ aptget update: Actualiza la base de datos local de paquetes.

→ aptget upgrade: Actualiza todos los paquetes instalados que pueda (solo si hay versiones nuevas disponibles)

- apt-get dist-upgrade: Actualiza todos los paquetes que pueda incluso cuando la actualización implique la instalación de paquetes nuevos.
- apt-get install paquete1 paquete2...: Instala paquetes apt-get
- apt-get remove [purge] paquete1 paquete2 ... : Desinstala paquetes.
- apt-get clean: elimina archivos descargados en /var/cache/apt/archives (para liberar espacios, NO desinstala los paquetes).

20. PAQUETES RPM/DEB

Paquetes .rpm

- Es un solo comando, dicho comando es usado para:
 - Instalar paquetes
 - Actualizar paquetes
 - Remover y desinstalar paquetes
 - Consultar la base de datos RPM
 - Verificar el paquete
 - Chequear archivos instalados
 - Construir un paquete binario desde los fuentes
- Chequea las siguientes cosas:
 - Verificación de espacio en disco suficiente para la instalación
 - Verifica que los archivos existentes no sean sobrescritos
 - Verifica que se cumplan todas las dependencias

| <i>Comando</i> | <i>Descripción</i> |
|------------------------|--|
| rpm -i paquete | Instala un paquete |
| rpm -e paquete | Desinstala un paquete |
| rpm -U paquete | Actualiza un paquete ya instalado |
| rpm -qi paquete | Nos da información de un paquete instalado |
| rpm -ql paquete | Vemos los ejecutables y demás archivos que lo acompañan |
| rpm -qpi paquete | Vemos su contenido sin necesidad de instalar |
| rpm -qa less | Vemos los paquetes rpm que tenemos instalados, con el “less” se pagina |
| rpm -qa more | Igual que el anterior |
| rpm -qa grep paquete | Busca algún paquete de entre los instalados que tenga paquete en su nombre |
| rpm -qil nombre | Para consultar donde está instalado |
| rpm -qf archivo | Te dice a que paquete pertenece un determinado archivo, pero hay que escribir la ruta completa de donde se encuentra el archivo. |

Paquetes .deb

DPKG, Sistema de gestión de paquetes de Debian

- Permite la instalación, borrado y mantenimiento de .debs
- dpkg i paquete.deb: instala un paquete.
- dpkg r [purge] paquete: elimina un paquete
- dpkg L paquete: muestra el contenido completo de un paquete.
- dpkg S file: busca paquetes que contengan el fichero.
- dpkg l: muestra la lista completa de paquetes instalados en el sistema.
- dpkg s paquete: muestra información del estado de un paquete.

Paquetes .deb

APT, front-end avanzado para DPKG

- APT resuelve dependencias. Al instalar un paquete puede que éste dependa de otro u otros para su funcionamiento. APT detecta esta dependencia e instala los paquetes necesarios.

- **Funcionamiento:**

- Se crea una base de datos local con la información de los paquetes instalables. Para crear esta base de datos hace falta un fichero con las fuentes (/etc/apt/sources.list) de donde bajarse la información.

- Esta base de datos local hay que actualizarla periódicamente (aptget update)

- Cuando se solicita la instalación de un paquete, APT comprueba primero en el sistema que el paquete no esté ya instalado y posteriormente comprueba en la base de datos local si el paquete está disponible.

- Si el paquete está disponible entonces se conecta a la fuente en cuestión para bajarse el paquete (archivo .deb). Estos paquetes bajados se guardan en /var/cache/apt/archives

- Posteriormente se procede a la instalación y configuración automática del paquete. Si el paquete necesita datos de configuración nos los pedirá el APT.

- **Definición de fuentes para APT:**

- Se definen en el fichero /etc/apt/sources.list

- Se pueden configurar con un asistente ejecutando aptsetup.

- Para leer CDs con fuentes podemos utilizar también aptcdrom.

- **Ejemplo de sources.list básico:**

```
deb http://ftp.fi.debian.org/debian stable main contrib nonfree
```

```
deb http://ftp.se.debian.org/debiannonUS stable/nonUS main contrib
```

```
aptcache: realiza búsquedas sobre la base de datos local de paquetes.
```

- **Útil para la obtención de información sobre software disponible.**

- aptcache

search patron: busca paquetes que cumplan un patrón.

- aptcache show paquete: muestra la información de un paquete

- aptcache depends paquete: muestra las dependencias del paquete.

- aptcache rdepends paquete: muestra las dependencias inversas del paquete.

ENTORNO GRÁFICO.

Entornos de ventanas en el ambiente gráfico

- Servidor X-Window
- Entornos de Escritorio
 - Kde
 - Gnome
 - Window Maker
 - IceWM
- Display Manager

La interfaz gráfica de usuarios se divide en 3 componentes:

- El servidor X-Window conocido como x ó x11

Ofrece operaciones gráficas básicas como abrir ventanas, visualizar imágenes, proporciona acceso a dispositivos como el teclado, mouse y monitor

El administrador de ventanas gestiona las operaciones de ventanas, como redimensionar o mover ventanas, los administradores de ventana difieren en el modo en que se presentan las ventanas utilizando diferentes bordes y menús de ventanas, todos utilizan el servidor X.

Conjunto de herramientas de interfaz de usuario: permite el acceso a archivos o programas en forma de iconos, menu de aplicaciones

EL FUNCIONAMIENTO DE X WINDOW

Mientras que el corazón de cualquier distribución es el kernel, para muchos usuarios, la cara del sistema operativo es el entorno gráfico proporcionando por el Sistema X Window, también llamado simplemente X.