



**Centro de Estudios Tecnológicos Industrial y de Servicios
No.115
Mexicano-Japonés**

MANUAL DE LINUX

Integrantes:

Duarte Gallegos Martha Angélica
Galera Delgado Eugenia Carolina
Godinez Vasquez Víctor Hugo

Materia:

Sistemas Operativos

miércoles 25 de abril de 2012

Indice

<i>Historia de Linux/Debian.....</i>	<i>3</i>
<i>Antecedentes de Linux/Debian.....</i>	<i>4</i>
<i>Debian se caracteriza por:.....</i>	<i>4</i>
<i>¿Que es Ubuntu?.....</i>	<i>5</i>
<i>Antecedentes de Ubuntu.....</i>	<i>5</i>
<i>¿Que proyección tiene ubuntu en el software libre?.....</i>	<i>7</i>
<i>¿Que proyección tiene Debian en el software libre?.....</i>	<i>7</i>
<i>Comandos de LINUX :.....</i>	<i>8</i>
<i>Estructura y archivos que usa Linux.....</i>	<i>21</i>

Historia de Linux/Debian

El proyecto Debian se creó de Agosto a Diciembre de 1993, con la versión 0.1-0.9, la que dio a conocer su creador Ian Murdock, el cual nombró al proyecto Debian por una fusión de su nombre y el de su mujer (Debra + Ian).

El proyecto Debian trabajó con el apoyo de Richard Stallman que, es el creador del proyecto GNU el cual se fundó en 1984 con el objetivo de crear un sistema operativo libre GNU.

Debian creció lentamente lanzando sus primeras versiones 0.9x y 1.x entre 1994 y 1996.

En el año de 1996, el creador de Debian fue sustituido por Bruce Perens a sugerencia del desarrollador Ean Schuessler, Bruce dirigió el Contrato Social de Debian y las pautas del software de Debian, definiendo puntos fundamentales para el desarrollo de la distribución. Hasta su retiro en 1998, cuando antes lanzó la primera versión de Debian basada en glibc, la 2.0

El proyecto procedió a elegir a nuevos líderes y a hacer revisiones de la versión 2.x, utilizando más versiones para otras arquitecturas y paquetes. Tiempo después fue lanzada como la primera portabilidad utilizando el núcleo de Hurd (del proyecto GNU) y no del núcleo de Linux como los anteriores; naciendo así Debian GNU/Hurd.

A finales de 2000, Debian realizó el mayor cambio a la estructura de los archivos y de las versiones, reorganizando procesos de liberación de paquetes del software con el nuevo "package pools", y creando una rama de prueba, estable para el próximo lanzamiento.

Desarrolladores comenzaron a reunirse en una conferencia anual llamada Debconf con discusiones y talleres para desarrolladores y usuarios técnicos.

Antecedentes de Linux/Debian

Debian se creó como una distribución de Linux completamente nueva. Cambiando la forma de estar desarrollada por un individuo aislado o un grupo, por desarrollarla abiertamente en el espíritu de Linux y GNU.

Su propósito principal de Debian es crear una distribución que esté a la altura del nombre de Linux. Debian se está ensamblando con cuidado y a conciencia, de tal manera que se le de un apoyo y mantenimiento con una atención similar.

Las distribuciones son esenciales para el futuro de Linux. En esencia, le eliminan al usuario la necesidad de buscar, obtener, compilar, instalar e integrar correctamente gran número de herramientas esenciales para conseguir un sistema Linux en funcionamiento.

En su lugar, construir el sistema sobre el creador de la distribución, con la intención de que muchos usuarios continúen usando una distribución por conveniencia incluso después de haberse familiarizado con el sistema operativo. De esta manera, las distribuciones juegan un papel realmente importante.

Debian se caracteriza por:

Tener disponibilidad en varias arquitecturas. (La versión estable incluye soporte para 12 plataformas):

- i386 – x86-32
- amd64 – x86-64
- alpha – DEC Alpha
- sparc – Sun SPARC
- arm – Arquitectura ARM
- armel – Emulador de ARM Emulator
- powerpc – Arquitectura PowerPC
- ia64 – Arquitectura Intel Itanium (IA-64)
- mips, mipsel – Arquitectura MIPS (big-endian y little-endian)
- s390 – Arquitectura IBM ESA/390 y z/Architecture
- m68k – Arquitectura Motorola 68k en Amiga, Atari, Mac, y varios

Sistemas embebidos VME

- Tener herramientas para facilitar el proceso de instalación y actualización del software
- Comprometerse con los principios y valores involucrados en el movimiento del Software Libre.
- No tiene ningún entorno gráfico en especial.
- Debian viene con más de 29000 elementos de software diferentes.
- Comprometerse con el contrato social de Debian y La Constitución de Debian.

¿Que es Ubuntu?

Ubuntu es un sistema operativo principalmente enfocado en la facilidad de uso e instalación, la libertad de los usuarios y los lanzamientos regulares(cada 6 meses). Esta sostenido por la empresa Canonical Ltd. y su comunidad de desarrolladores. Utiliza un núcleo Linux y esta basado en Debian GNU/Linux . Esta constituido por múltiples software de licencia libre o código abierto. Ubuntu esta diseñado principalmente para su uso en ordenadores personales aunque también proporciona soporte para servidores.

Su nombre proviene del concepto africano ubuntu, que significa "humanidad hacia otros" o "yo soy porque nosotros somos".

Antecedentes de Ubuntu

El 8 de julio de 2004, Mark Shuttleworth y la empresa Canonical Ltd. anunciaron la creación de la distribución Ubuntu. Ésta fue financiada inicialmente con 10 millones de dólares (US\$). Este proyecto surgió por iniciativa de algunos programadores de los proyectos Debian, Gnome que no estaban contentos con la manera de operar del proyecto Debian.

Según los fundadores de Ubuntu, Debian era un proyecto demasiado burocrático donde no había responsabilidades definidas y donde cualquier propuesta interesante se ponía en discusión y nunca se llegaba a un acuerdo.

Al no sentirse contentos y después de formar un grupo multidisciplinario, los programadores buscaron el apoyo económico de Mark Shuttleworth, un emprendedor sudafricano.

Y así nació la empresa Canonical, la cual se encarga de sostener económicamente el proyecto mediante la comercialización de servicios y soporte técnico a otras empresas. Después de varios meses de trabajo y un periodo de pruebas, fue lanzada la primera versión de Ubuntu (Warty Warthog) el 20 de octubre de 2004.

Cada lanzamiento de Ubuntu posee un nombre en clave, como también un número de versión basado en el año y el mes del lanzamiento. Por ejemplo, la versión 4.10 fue lanzada en octubre (10) de 2004 (4). Cada versión de Ubuntu es lanzada con seis meses de diferencia con respecto al último lanzamiento.

La última versión estable de Ubuntu estable es la **10.04**.

Variantes

- Kubuntu
- Edubuntu
- Xubuntu
- Ubuntu Studio

¿Que proyección tiene ubuntu en el software libre?

El impacto que ha tenido ubuntu en el software libre así es importante ya que se ha logrado que los usuarios se interesen por el software libre y poder hacer crecer esta gran comunidad. Ubuntu ahora ya lo podemos encontrar en computadoras, servidores, Tablets, Smart TVs y Smartphones y recientemente se anunció que está disponible para TV (Ubuntu TV) y para Android (Ubuntu for Android), el cual permite ejecutar el escritorio de Ubuntu directamente desde un smartphone Android al conectarse en un monitor.

Tanto ha sido el éxito del software libre que cada vez más se va ampliando su distribución y las personas empiezan a conocer otros sistemas operativos diferentes del clásico Windows. Al suceder esto el software libre será haciendo más común y podrá ir ampliando, mejorando y se podrán ir creando nuevas aplicaciones ...Así que hay que fomentar el uso del software libre!!

¿Que proyección tiene Debian en el software libre?

Aunque no hay disponibles estadísticas precisas (ya que Debian no requiere que los usuarios se registren), hay signos bastante evidentes de que Debian lo usan un amplio número de organizaciones, grandes y pequeñas, así como muchos miles de personas de forma individual. Debian así como sus derivados son usados actualmente en organizaciones gubernamentales, comerciales, instituciones educativas y organizaciones sin fines de lucro. Esto se debe a su seguridad, su facilidad de uso y su popularidad, así como muchas personas también lo usan y eso sucede en gran parte del mundo, también hay que recordar que Ubuntu está basado en Debian, se podría decir que es un derivado de Debian.

Así como también se usa en computadoras y servidores. Se tiene que seguir impulsando el uso del software libre contribuir a que esto siga adelante.

El movimiento del software libre comenzó en 1983 cuando Richard Stallman anunció el proyecto GNU. La meta del movimiento fue dar libertad a los usuarios de computadoras reemplazando el software con términos de licencia restrictivos, como el software propietario, por software libre.

Comandos de LINUX :

Un intérprete de comandos es un programa que toma la entrada del usuario, por ejemplo las órdenes

que teclea, y la traduce a instrucciones.

Para pasar al modo texto (intérprete de comandos) desde el modo gráfico hemos de pulsar las

combinaciones: Ctrl + Alt + F1 o bien con: F2 F3 F4 F5 F6

Esto hace que el sistema salga del modo gráfico y acceda a alguna de las seis consolas virtuales de

Linux, a las cuales también se puede acceder cuando se arranca en modo de texto.

Para volver al modo gráfico hay que presionar Ctrl + Alt + F7 o Ctrl + Alt + F8 (Según la sesión en

modo gráfico a la que deseemos regresar).

En una terminal:

Las aplicaciones con nombres compuestos se escriben con guión entre las palabras (ej. compizconfig-settings-manager).

Para los espacios en blanco se pone un guión inferior (ej. nombre_archivo)

Cuando queremos poner varios comandos seguidos, para ejecutarlos a la vez, dejamos un espacio

entre ellos, nunca una coma (ej. avidemux k3b kde-i18n-es k3b-i18n).

cd:

cd se utiliza para cambiar directorio y es por su significado en ingles que es change directory

cd .. se utiliza para retroceder un directorio.

Cd /mas el nombre a donde quieras entrar/ te lleva a la direccion solicitada.

Cd /el nombre de donde quieras entrar /nombre de lo que quieras buscar/ te lleva a una pagina mas especifica .

mk:

crea un directorio vacío en el lugar de disco en donde te encuentres este comando tiene mas propiedades algunas se mencionan a continuacion

mkdir -m[permisos_octal] directorio crea una carpeta donde se le pueden poner permisos.

mkdir -p directorio/subdirectorio crea un directorio pero si no hay directorios "padre" tambien los crea.

mkdir -v directorio. Muestra un mensaje a pantalla diciendo que el directorio ha sido creado.

rm:

se utiliza para borrar archivos o carpetas cuando se usa este comando ya no se puede deshacer para este pueden aver distintas formas de usarlo

rm archivo --> borra el archivo pero no puede borrar directorios

rm -f archivo --> no pide que confirmes si quieres borrar o no y los borra.
rm -i archivo --> este pide la confirmación para borrar cada archivo.
rm -r directorio --> borra todo lo que contiene e directorio.

ls:

Es el mas usado en Linux y se usa para listar directorios y subdirectorios tiene muchas opciones; ls tiene mas propiedades como por ejemplo:
ls -l aparte de listar directorios tambien lista propiedades.
ls -a lista directorios aunque esten ocultos.
Estos son solo dos ejemplos de el los diferentes usos de ls.

aptitude:

aptitude sirve para instalar programas con programas secundarios que de alguna manera te ayudan con lo que le pediste hacer y también desinstalan programas que el mismo instalado.

aptitude update = Actualiza la lista de paquetes.
aptitude upgrade = Actualiza el sistema (no instala ni elimina paquetes).
aptitude dist-upgrade = Actualiza el sistema eliminando e instalando paquetes si fuera necesario.
aptitude install [paquetes] = Instala los paquetes indicados.
aptitude reinstall [paquetes] = Reinstala los paquetes indicados.
aptitude remove [paquetes] = Elimina los paquetes indicados.
aptitude purge [paquetes] = Elimina los paquetes y sus ficheros de configuracion.
aptitude download [paquetes] = Descarga los paquetes en el directorio actual.
aptitude hold [paquetes] = Bloqua los paquetes indicados.
aptitude unhold [paquetes] = Desbloquea los paquetes seleccionados.
aptitude unmarkauto [paquetes] = Desmarca paquetes como instalados manualmente.
markauto = Marca paquetes como instalados manualmente.
aptitude search [expresion] = Busca un paquete por nombre o expresion.
aptitude show [paquete] = Muestra informacion detallada de un paquete.
aptitude clean = Elimina los paquetes .deb descargados.

apt -get:

se utilizan para las distribuciones y derivados en Debian , actualiza e instala programas remotos de paquetes en sistemas basados en linux debian.

apt-get update

Actualiza la lista de paquetes, de todos repositorios que tengamos.

apt-get upgrade

Actualiza todos los paquetes que tengan actualización, tanto si son del sistema como si no.

apt-get autoremove

Elimina paquetes que han quedado huérfanos tras alguna desinstalación de algún paquete.

apt-cache search nombre_paquete

Busca nombre_paquete en la lista de paquetes y te muestra todos los que encuentre que contengan nombre_paquete.

apt-get install nombre_paquete

Descarga el paquete llamado nombre_paquete y sus dependencias, tras lo cual los instala.

apt-get remove [--purge] nombre_paquete

Elimina el paquete llamado nombre_paquete. Con el argumento "--purge" borra también los archivos de configuración.

find:

El comando find encuentra uno o más archivos asumiendo que conoces su nombre aproximado.

La sintaxis es:

find ruta [opciones]

Opciones:

-name	Busca dicho archivo, en el directorio actual o en cualquier subdirectorio.
-atime n	Verdadero si se accedió al archivo hace n días. El tiempo de acceso de los directorios en la ruta se cambia por el find mismo.
-ctime n	Verdadero si el estado del archivo se cambió hace n días.
-group gname	Verdadero si el archivo pertenece al grupo gname. Si gname es numérico y no aparece en el archivo /etc/group, se toma como identificador de grupo.
-mtime n	Verdadero si la información del archivo fue modificada hace n días.
-size n[c]	Verdadero si el archivo tiene n bloques de largo (512 octetos por bloque). Si n va seguida de una c, el tamaño es en octetos.

Ejemplo:

find -name 'cal.txt'

El sistema buscará cualquier archivo llamado "cal.txt" en el directorio actual y en cualquier subdirectorio.

Tar:

se utiliza para comprimir y descomprimir archivos.

tar -vcf archivo.tar /fichero1 /fichero2 ... = Empaqueta ficheros o directorios en .tar

tar -vxf archivo.tar = Desempaqueta el archivo .tar

tar -vtf archivo.tar = Muestra el contenido del archivo .tar sin descomprimirlo.

tar -zvcf archivo.tgz directorio = Empaqueta y comprime (tgz) directorios o ficheros.

tar -zvxf archivo.tgz = Desempaqueta y descomprime un archivo.tgz

tar -zvtf archivo.tgz = Muestra el contenido de un tgz sin descomprimirlo ni desempaquetarlo.

tar -jvcf archivo.tbz2 directorio = Empaqueta y comprime (tbz2) directorios o ficheros.

tar -jvxf archivo.tbz2 = Desempaqueta y descomprime el archivo.tbz2

tar -jvtf archivo.tbz2 = Muestra el contenido sin desempaquetar ni descomprimir el .tbz2

du:

El comando du se usa para informar de cuánto espacio en disco ocupa un archivo o directorio.

La sintaxis es:

du [opciones] directorios

Opciones:

- a Muestra el uso de espacio de cada archivo.
- k Escribe el tamaño de los archivos en unidades de 1024 octetos, en vez de las unidades de 512 octetos por defecto.
- s En vez de el salida por defecto, informa sólo de la suma total de cada uno de los archivos especificados.
- L Procesa los enlaces simbólicos usando el archivo o directorio al que el enlace simbólico referencia, en vez de el enlace en sí mismo.
- x Cuando evalúa los tamaños de archivo, evalúa sólo esos archivos que tengan el mismo dispositivo que el archivo especificado en el operando.

Ejemplo:

du -h

Salida:

84K

Muestra la capacidad de la carpeta actual.

df:

El comando df se usa para informar de cuánto espacio libre en disco está disponible para cada partición que tengas. La primera columna muestra el nombre de la partición tal como aparece en el directorio /dev. Las columnas siguientes muestran el espacio total, bloques asignados y bloques disponibles.

La sintaxis es:

df [opciones]

Opciones:

- a Incluye sistemas de archivos falsos.
- h Mostrar los tamaños en formato legible por humanos (1K 234M 2G)
- H Muestra tamaños en formato legible por humanos, pero utiliza potencias de 1000, no de 1024.
- i Listar información de inodos en vez de uso de bloques.
- l Limitar el listado a sistemas de archivos locales.
- P Usar el formato de salida POSIX.
- T Mostrar el tipo de sistema de archivos.

Ejemplo:

df -h

Salida:

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
                144G  14G  123G  11% /
/dev/sda1       99M   11M   83M  12% /boot
tmpfs           248M   0   248M   0% /dev/shm
```

Mostrar tamaños en formato legible por humanos.

fdisk:

El comando fdisk se usa para manipular la tabla de particiones. Los discos duros pueden dividirse en uno o más discos lógicos llamados particiones.

La sintaxis es:

fdisk [opciones]

Opciones:

- l Enlista las tablas de partición para los devies especificados y sale.
- u Cuando enlista tablas de partición, muestra tamaños en sectores en vez de cilindros.
- s El tamaño de la partición se muestra en el salida estándar.
- b Especifica el tamaño de sector del disco.
- C Especifica el número de cilindros del disco.
- H Especifica el número de cabezas del disco.
- S Especifica el número de sectores por pista del disco.

Ejemplo:

```
fdisk /dev/hdb
```

Muestra la tabla de partición y la información de configuración.

dpkg

Despaquetar

Los paquetes cuando se instalan sufren un proceso de despaquetaje. En el fondo un paquete .deb

contiene una serie de scripts de pre-instalación, post-instalación y los archivos en cuestión del paquete.

Este comando lo usaremos para instalar un paquete .deb que ya tengamos descargados en nuestro sistema.

Si queremos instalar un paquete ya descargado mediante consola usaremos el argumento '-i' (i=install):

`dpkg -i paquete` = Instala un paquete.
`dpkg -r paquete` = Desinstala un paquete.
`dpkg --purge paquete` = Desinstala un paquete y sus archivos de configuración.
`dpkg --force -r paquete` = Fuerza la desinstalación de un paquete.
`dpkg --force-all -r paquete` = Fuerza aún más la desinstalación de un paquete (Puede comprometer el sistema).
`dpkg -c paquete` = Muestra el contenido de un paquete.
`dpkg -L paquete` = Muestra todos los ficheros que se instalaron con un paquete.
`dpkg -S fichero` = Muestra a qué paquete pertenece un fichero.
`dpkg --get-selections` = Muestra un listado con todos los paquetes instalados.
`dpkg-reconfigure paquete` = Reconfigura el paquete.

grep:

Su funcionalidad es la de escribir en salida estándar aquellas líneas que concuerden con un patrón. Su sintaxis es como sigue:

```
grep [opciones] PATRÓN [ARCHIVO...]
```

```
grep [opciones] [-e PATRÓN | -f ARCHIVO] [ARCHIVO...]
```

Este comando realiza una búsqueda en los ARCHIVOS (o en la entrada estándar, si no se especifica ninguno) para encontrar líneas que concuerden con PATRÓN. Por defecto `grep` imprime en pantalla dichas líneas. Sus opciones más interesantes son:

Opciones:

-c

Modificar la salida normal del programa, en lugar de imprimir por salida estándar las líneas coincidentes, imprime la cantidad de líneas que coincidieron en cada archivo.

-e PATRÓN

Usar PATRÓN como el patrón de búsqueda, muy útil para proteger aquellos patrones de búsqueda que comienzan con el signo «-».

-f ARCHIVO

Obtenga los patrones del archivo ARCHIVO

-H

Imprimir el nombre del archivo con cada coincidencia.

-r

Buscar recursivamente dentro de todos los subdirectorios del directorio actual.

El patrón de búsqueda normalmente es una palabra o una parte de una palabra. También se pueden utilizar expresiones regulares, para realizar búsquedas más flexibles.

Ejemplo 11. Algunos ejemplos simples con el comando `grep`

Si se quisiera buscar la ocurrencia de todas las palabras que comiencen con «a» minúscula, la ejecución del comando sería algo así:

```
$ grep 'a*' archivo
```

chown:

El comando `chown` se usa para cambiar el propietario / usuario del archivo o directorio. Es un comando de administrador, sólo el usuario `root` puede cambiar el propietario de un archivo o directorio.

La sintaxis es:

```
chown [opciones] nuevo_usuario nombre_de_archivo/directorio
```

Opciones:

- R Cambia el permiso en archivos que estén en subdirectorios del directorio en el que estés en ese momento.
- c Cambia el permiso para cada archivo.
- f Previene a `chown` de mostrar mensajes de error cuando es incapaz de cambiar la titularidad de un archivo.

Ejemplo:

```
chown hiox test.txt
```

El dueño del archivo "test.txt" es `root`, cambia al nuevo usuario `hiox`.

chmod:

El comando `chmod` te permite alterar / cambiar los derechos de acceso a archivos y directorios.

La sintaxis es:

```
chmod [options] [MODE] FileName
```

Permiso de Archivo

#	Permiso de Archivo
0	ninguno
1	sólo ejecutar
2	sólo escritura
3	escritura y ejecución
4	sólo lectura
5	lectura y ejecución
6	lectura y escritura

7 todos los permisos

Opciones:

- c Muestra los nombres de aquellos archivos cuyos permisos estén siendo cambiados
- f Elimina la mayoría de los mensajes de error
- R Cambia archivos y directorios repetidamente
- v Mostrar la información de la versión y salir.

Ejemplo:

Para ver qué permisos tienen tus archivos:

ls -alt

update-grup:

Este comando buscará los kernels instalados en las particiones y después añadirá unas entradas genéricas al final del archivo menu.lst - dichas entradas deberían de permitir arrancar los kernels instalados - pero pueden ser necesarios ajustes adicionales para que se amolden al caso concreto de cada sistema.

sudo:

La orden sudo permite ejecutar una orden como otro usuario (obtenemos sus permisos), normalmente se usa para obtener los permisos de Root.

Sintaxis básica

La sintaxis es:

sudo [opciones] [orden]

Descripción de opciones principales con algunos ejemplos simples

-: A continuación pedirá el password de usuario Root, así establecemos que todo lo que se haga en adelante será hecho por el super usuario.

-h: (help/ayuda) muestra un mensaje de ayuda con los usos de sudo.

-L: Con esta opción se nos presenta una lista con las Opciones de Configuración (Defaults) de /etc/sudoers con una corta descripción de cada una de ellas.

-l: Nos muestra una relación de los comandos que el usuario puede ejecutar en el presente host.

-V: muestra la versión de sudo

-k: Esta opción no requiere contraseña y invalida el tiempo de vigencia de la contraseña del

usuario.

-K: Análoga a la opción -k, con la diferencia que elimina completamente el tiempo de vigencia de la contraseña del usuario para la sesión.

sudo passwd = Permite al usuario cambiar o crear la contraseña de root.

Ejemplo:

```
sudo aptitude install emacs
```

more:

El comando more se usa para mostrar texto en la pantalla del terminal. Muestra un fichero de forma tabulada como una pagina del man.

La sintaxis es:

```
more [opciones] nombre_archivo
```

Opciones:

- c Limpia la pantalla antes de mostrar.
- e Salir inmediatamente después de escribir la última línea del último archivo en la lista de argumentos.
- n Especifica cuántas líneas se muestran en la pantalla para un archivo dado.
- +n Inicia el archivo desde el número dad

Ejemplo:

```
more -c index.php
```

Limpia la pantalla antes de mostrar el archivo.

tail:

El comando tail se usa para mostrar la parte final de un archivo. Por defecto muestra las últimas 10 líneas de un archivo.

La sintaxis es:

```
tail [opciones] nombre_de_archivo
```

Opciones:

- l Especifica las unidades de líneas.
- b Especifica las unidades de bloques.
- n Especifica cuántas líneas quieres mostrar.
- c número El número debe ser un entero decimal cuyo signo afecte a la localización en el archivo, medido en bytes.
- n número El número debe ser un entero decimal cuyo signo afecte a la localización en el archivo, medido en líneas.

Ejemplo:

tail index.php

Muestra las últimas 10 líneas de "index.php"

head:

Muestra las primeras líneas del contenido de los archivos especificados; por defecto muestra las 10 primeras líneas.

La sintaxis es:

head [opciones] nombre_de_archivo

Opciones:

- n Especifica cuántas líneas quieres mostrar.
- n número El número debe ser un entero decimal cuyo signo afecte a la localización en el archivo, medido en líneas.
- c número El número debe ser un entero decimal cuyo signo afecte a la localización en el archivo, medido en octetos.

Ejemplos:

head index.php

Este comando muestra las primeras 10 líneas de "index.php".

head -5 index.php

El comando head muestra las primeras 5 líneas de "index.php".

wc:

El nombre del comando wc proviene de word count, y como es de suponer, sirve para contar palabras. Pero no sólo palabras como se verá a continuación. Su sintaxis es como sigue:

wc [opción...] [archivo...]

Si se omite el argumento archivo, wc tomará los datos (naturalmente) de la entrada estándar.

La lista de opciones más importantes es la siguiente:

-c
Contar bytes.

-l
Contar líneas.

-w
Contar palabras.

Ejemplo:

Como ejemplo, se pueden contar las líneas del archivo `/etc/passwd` y de esta manera se sabrá rápidamente cuántos usuarios tiene definidos el sistema:

```
~$ wc -l /etc/passwd
  32 /etc/passwd
```

Se pueden combinar varios argumentos a la vez.

vi:

Es un clásico editor en ambientes tipo UNIX cuya prioridad es la eficiencia. Por lo mismo no requiere el uso de ratón sino de breves comandos que a medida que se memoricen y usen harán más rápida la edición de textos.

Para editar un archivo (digamos `ma.txt`) con el editor `vi`, teclee desde un intérprete de comandos:

```
~$ vi ma.txt
```

nano:

Editor de texto; lee, crea, edita un fichero de texto.

M = Esc/Alt | | Ctrl + g ----- Invocar el menú de ayuda.
Ctrl + x ----- Salir de nano.
Ctrl + o ----- Escribir el fichero actual a disco.
Ctrl + r ----- Insertar otro fichero en el actual.
Ctrl + w ----- Buscar un texto en el editor.
Ctrl + y ----- Moverse a la página anterior.
Ctrl + v ----- Moverse a la página siguiente.
Ctrl + k ----- Cortar la línea actual y guardarla en el cutbuffer.
Ctrl + u ----- Pegar el cutbuffer en la línea actual.
Ctrl + l ----- Redibujar la pantalla actual.
Ctrl + j ----- Justificar el párrafo actual.
Ctrl + m ----- Insertar un retorno de carro en la posición del cursor.
Ctrl + _ ----- Ir a un número de línea en concreto.
M + g ----- Ir a un número de línea en concreto.
M + i ----- Auto indentar habilitar/deshabilitar.
M + x ----- Modo ayuda habilitar/deshabilitar.
M + p ----- Modo Pico habilitar/deshabilitar.
M + m ----- Soporte para ratón habilitar/deshabilitar.
M + r ----- Reemplazar texto en el editor.
M + e ----- Expresiones regulares habilitar/deshabilitar.

M + b ----- Respalda ficheros habilitar/deshabilitar.
M + s ----- Desplazamiento suave habilitar/deshabilitar.
M + h ----- Tecla '\smart home\' habilitar/deshabilitar.
M + y ----- Coloreado de sintaxis habilitar/deshabilitar.
M + p ----- Mostrar blancos habilitar/deshabilitar.

emacs:

Emacs es un editor de texto con una gran cantidad de funciones.
Sus comandos son:
<http://www.gfc.edu.co/~andpin/comandos.html>

gcc:

GCC es un compilador integrado del proyecto GNU para C, C++, Objective C y Fortran; es capaz de recibir un programa fuente en cualquiera de estos lenguajes y generar un programa ejecutable binario en el lenguaje de la máquina donde ha de correr.
La sigla GCC significa "GNU Compiler Collection". Originalmente significaba "GNU C Compiler"; todavía se usa GCC para designar una compilación en C. G++ refiere a una compilación en C++.

Sintaxis:

gcc [opción | archivo] ...
g++ [opción | archivo] ...

Las opciones van precedidas de un guión, como es habitual en UNIX, pero las opciones en sí pueden tener varias letras; no pueden agruparse varias opciones tras un mismo guión. Algunas opciones requieren después un nombre de archivo o directorio, otras no. Finalmente, pueden darse varios nombres de archivo a incluir en el proceso de compilación.

gcc -o hola hola.c
compila el programa en C hola.c, gener un archivo ejecutable hola.
g++ -o hola hola.cpp
compila el programa en C++ hola.c, gener un archivo ejecutable hola.

Sufijos en nombres de archivo.

Son habituales las siguientes extensiones o sufijos de los nombres de archivo:

.c	fuelle en C
.C .cc .cpp .c++ .cp .cxx	fuelle en C++; se recomienda.cpp
.m	fuelle en Objective-C
.i	C preprocesado
.ii	C++ preprocesdo
.s	fuelle en lenguaje ensamblador
.o	código objeto
.h	archivo para preprocesador (encabezados), no suele figurar en la línea de c

Opciones.

- c

realiza preprocesamiento y compilación, obteniendo el archivo en código objeto; no realiza el enlazado.

- E

realiza solamente el preprocesamiento, enviando el resultado a la salida estándar.

-o *archivo*

indica el nombre del archivo de salida, cualesquiera sean las etapas cumplidas.

-I*ruta*

especifica la ruta hacia el directorio donde se encuentran los archivos marcados para incluir en el programa fuente. No lleva espacio entre la I y la ruta, así: -I/usr/include

-L

especifica la ruta hacia el directorio donde se encuentran los archivos de biblioteca con el código objeto de las funciones referenciadas en el programa fuente. No lleva espacio entre la L y la ruta, así: -L/usr/lib

-Wall

muestra todos los mensajes de error y advertencia del compilador, incluso algunos cuestionables pero en definitiva fáciles de evitar escribiendo el código con cuidado.

-g

incluye en el ejecutable generado la información necesaria para poder rastrear los errores usando un depurador, tal como GDB (GNU Debugger).

-v

muestra los comandos ejecutados en cada etapa de compilación y la versión del compilador. Es un informe muy detallado.

Mas en:

<http://iie.fing.edu.uy/~vagonbar/gcc-make/gcc.htm>

grub:

GRUB (GRand Unifier Bootloader) es un gestor de arranque: es lo primero que se carga cuando se inicia la computadora. Permite tener diferentes sistemas operativos, y diferentes versiones de ellos, en el mismo disco duro. Por ejemplo podemos tener Windows y GNU/Linux en la misma computadora, GRUB se cargará antes que cualquiera de éstos permitiéndonos elegir cuál iniciar.

GRUB permite varios comandos en su línea de comandos.

Ejemplo.

Para modificar el tiempo de espera, sistema operativo por defecto, el nombre de los sistemas operativos y toda la información del arranque de cada uno de ellos en /boot/grub/menu.ls

Estructura y archivos que usa Linux

Es importante conocer la estructura interna del sistema que utilicemos para poder tener control sobre las operaciones que realizamos sobre él. Además de familiarizarnos mejor con el entorno, resulta útil a la hora de buscar un archivo específico.

En Linux existen diversos directorios y subdirectorios que realizan esta labor. Librerías, binarios, programas instalados, archivos temporales, etc., que se pueden encontrar en sitios específicos donde podemos ubicarlos si necesitamos en algún momento de ellos.

Como por ejemplo algunos de ellos:

/

Directorio raíz donde se encuentran todos los directorios y archivos de una manera lógica.

/bin

En este directorio se puede encontrar todos los archivos ejecutables del sistema, así como muchos de los comandos, esencialmente como: arch, cat, chgrp, chmod, chown, cp, date, dd, df, dmesg, echo, ed, false, kill, in, login, mxdir, mknod, more, mount, mv, ps, pwd, rm, rmdir, sed, setser

/home

Aquí se encuentran todos los archivos de los clientes del sistema. Como por ejemplo: Si el nombre del cliente es Oliver sería /home/Oliver

/media

Generalmente aquí es donde se instalan los Cds y Dvds además de los SticksUSB y discos duros externos.

/sbin

Tiene archivos ejecutables que por lo general son comandos usados para la administración del sistema. Los comandos mount, halt, umount y shutdown son algunos de ellos.

/usr

Contiene varios archivos y subdirectorios importantes como pueden ser las configuraciones del entorno grafico X, fuentes del kernel, librerías, juegos, etc.

/boot

Todos los archivos necesarios para el arranque incluidos el cargador Grub y los kernels disponibles.

/mnt

Es un directorio vacío, normalmente suele usarse para montajes de unidades temporales que deseamos cargar en ese momento.

/var

Cuenta con varios archivos que definen el sistema así como archivos log que almacenan cada movimiento del sistema como por ejemplo el fichero /var/log/messages.

/cdrom

Normalmente es un enlace simbólico hacia /media/cdrom

/dev

Aquí es donde se encuentran todos los drivers y los dispositivos, estos se identifican en forma de archivo.

/lib

Contiene librerías para C y otros lenguajes de programación.

/proc

Directorio que contiene información sobre diferentes partes del sistema, cpu, discos, tiempo uptime, irqs, memoria, etc ...

/opt

En este directorio se suelen almacenar todos los archivos de una instalación fuera de los repositorios como puede ser cuando instalamos un .deb

/etc

Contiene prácticamente todos los archivos de configuración del equipo y los demonios de inicio en /etc/init.d, así como otros archivos por ejemplo: adjtime, csh.login, disktab, fdprm, fstab, gettydefs, group, inittab, issue, ld.so.conf, lilo.conf, magic, motd, mtab, mtools, passwd, profile, psdataba.

/lost+found

Directorio específico para archivos perdidos. Cada partición tiene el suyo propio independientemente.

/root

Directorio particular del superusuario del sistema (root). Si el directorio hogar de root no está almacenado en la partición raíz, será necesario asegurarse que tome / por defecto si no puede ser localizado.

/tmp

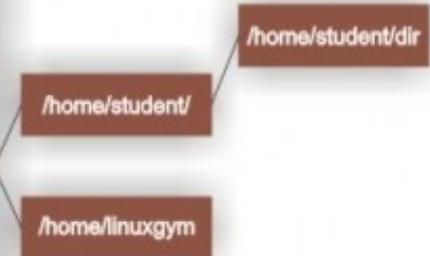
Directorio temporal que pueden usar todos los clientes para archivos temporales y del sistema.

Directorio raíz de toda la jerarquía de archivos del sistema

/

Jerarquía primaria

/bin/	Comandos binarios esenciales de usuario
/boot/	Archivos estáticos del selector de arranque
/dev/	Archivos de unidades
/etc/	Configuración de sistema de Host específico Directorios requeridos: opt, X11, sgml, xml
/home/	Directorio 'home' de usuario
/lib/	Librerías esenciales compartidas y módulos de Kernel
/media/	Punto de montaje para medios removibles
/mnt/	Punto de montaje temporal para sistemas de archivos
/opt/	Agregados de paquetes de Software y Aplicaciones
/sbin/	Binarios de sistema
/srv/	Datos para los servicios provistos por este sistema
/tmp/	Archivos temporales
/usr/	Utilidades y aplicaciones de (Multi-)usuario Jerarquía secundaria Directorios requeridos: bin, include, lib, local, sbin, share
/var/	VARIABLES de archivo
/root/	Directorio 'home' del usuario root
/proc/	Documentación del sistema de archivos virtual del Kernel y las condiciones de los procesos en archivos de texto



Jerarquía estándar de archivos del sistema



LINUXCONFIG.ORG

Traducido a español por Franco Ferrari
 franco.ferrari@rinconmovil.com
<http://eldebianita.hazlo-asi.biz/>