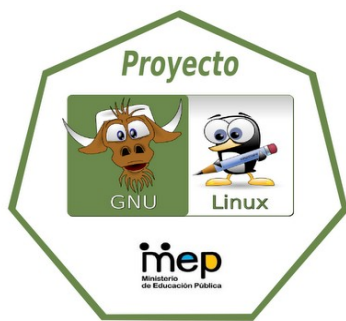


The system is go
linux-2dyx:/home/
linux-2dyx:/home/
linux-2dyx:/home/
linux-2dyx:/home/
linux-2dyx:/home/
linux-2dyx:/home/

Administración de Servidores **Ubuntu Server 11.4**

Lección #3: La consola, comandos de administración Ubuntu Server



En este capítulo veremos el funcionamiento del Shell, la Línea de comandos y el emulador de terminal mediante los comandos que GNU/Linux nos ofrece para comunicarnos con el ordenador. Para poder hacerlo debemos saber un poco el funcionamiento de los sistemas operativos.

Un computador por sí mismo no tiene funcionalidad, el hardware sin software no sirve más que para un pisapapeles. El Sistema Operativo (como lo son UNIX, GNU/Linux, FreeBSD, Solaris, etc) son el alma de los ordenadores. Los Sistemas Operativos están formados por varias partes: El kernel, el Shell y las aplicaciones.

El kernel es el encargado de comunicarse con el hardware. Él convierte las ordenes introducidas en el Shell a un conjunto de valores binarios que pasan por todos los componentes electrónicos y dan como resultado lo que el usuario quiere.

El Shell se encarga de interpretar las ordenes que vienen desde las aplicaciones, las convierte en un lenguaje que se encuentra entre lo que “entiende” la máquina y lo que entiende el usuario.

Las aplicaciones son las que reciben órdenes directamente de los usuarios, y muestran a ellos los resultados obtenidos de todo el proceso anteriormente explicado.

La línea de comandos es el método que se utiliza para dar órdenes a un programa informático mediante una única línea de código. Estás pasan por un intérprete de comandos, el cual es el programa encargado de “saber” qué orden se escribe en esa línea de código y pasarla al Shell. La comunicación se da en una sola dirección, del usuario a la máquina.

Por otra parte tenemos el Emulador de Terminal, el cual ejecuta una orden dada por el usuario, revisa su sintaxis (que en realidad sea una orden válida) y la ejecuta, pero, además de esto nos muestra el resultado en pantalla. Por eso mismo se dice que la comunicación entre el usuario y la máquina se da en ambas direcciones.

Es muy importante recalcar que en los sistemas basados en Unix, como lo es GNU/Linux, el sistema está conformado de Ficheros y Procesos. Los procesos son las acciones que se realizan en el computador, y los ficheros son los objetos sobre los cuales actúan los procesos. Desde un documento de texto hasta el lector óptico son ficheros.

En el desarrollo del capítulo se obtendrán conocimientos sobre el manejo de los tipos de usuarios, comandos de administración de estos y otros componentes del sistema operativo, diagnóstico del sistema, combinación de algunos comandos, etc.

Árbol de Directorios:

En GNU/Linux todo lo que se puede utilizar son ficheros y estos se organizan en directorios. Es importante conocer su contenido para mantener el orden y poder realizar cambios precisos en archivos de configuración, instalación de programas y cualquier otro menester. Al igual que todos los sistemas operativos basados en UNIX todo empieza en “/” a la cual llamaremos “raíz del sistema”, dentro de “/” se encuentran los otros directorios, cada uno con su función específica. La configuración básica del Árbol de Directorios (Conjunto de directorios principales) de GNU/Linux es la siguiente:

/	La raíz, que contiene los directorios principales
/bin	Contiene archivos ejecutables fundamentales del sistema, utilizados por todos los usuarios (como por ejemplo los comandos ls , rm , cp , chmod , mount , etc.).
/boot	Contiene los archivos que permiten que Linux se inicie
/dev	Contiene los puntos de entrada para los periféricos
/etc	Contiene los comandos y los archivos que el administrador del sistema necesita (archivos passwd , group , inittab , ld.so.conf , lilo.conf , etc.)
/etc/X11	Contiene los archivos específicos para la configuración de X (XF86Config , por ejemplo)
/etc/opt	Contiene los archivos de configuración específicos para las aplicaciones instaladas en /opt
/home	Directorio personal del usuario
/lib	Contiene bibliotecas compartidas que son fundamentales para el sistema durante su inicio
/mnt	Contiene puntos de montaje de particiones temporales (CD-ROM, disquete, etc.)
/opt	Contiene paquetes de aplicaciones suplementarias
/root	Directorio del administrador de raíz
/sbin	Contiene los sistemas binarios fundamentales (por ejemplo, el comando adduser)
/tmp	Contiene archivos temporales
/usr	Jerarquía secundaria
/usr/X11R6	Este directorio se reserva para el sistema X versión 11.6
/usr/X386	Éste es un vínculo simbólico con /usr/X11R6, el cual utilizó previamente X versión 5
/usr/bin	Contiene la mayor parte de los archivos binarios y los comandos del usuario
/usr/include	Contiene los archivos de encabezado para los programas C y C++
/usr/lib	Contiene la mayoría de las bibliotecas compartidas del sistema
/usr/local	Contiene datos que pertenecen a los programas instalados en la raíz del equipo local
/usr/local/bin	Binarios de programas locales
/usr/local/games	Binarios de juegos locales
/usr/local/include	Archivos de encabezado locales de C y C++
/usr/local/lib	Bibliotecas locales compartidas
/usr/local/sbin	Binarios del sistema local
/usr/local/share	Jerarquía independiente
/usr/local/src	Archivos fuente locales
/usr/sbin	Contiene los archivos binarios que no son fundamentales para el sistema y que se reservan para el administrador del sistema
/usr/share	Reservado para datos independientes de la arquitectura
/usr/src	Contiene archivos fuente de código
/var	Contiene datos variables

Fuente: <http://es.kioskea.net/contents/linux/>, gracias por su valioso aporte a la comunidad.

Usuarios en GNU/Linux, conceptos básicos

En informática, un usuario es un individuo que utiliza el ordenador, sistema operativo, servicio o cualquier sistema informático. GNU/Linux es un sistema multiusuario, en otras palabras, muchos usuarios pueden utilizar el sistema operativo al mismo tiempo.

Los usuarios en un sistema operativo se utilizan para restringir el acceso a ficheros y/o directorios, o para permitirles acceso a otros ficheros y/o directorios. Además se puede programar controles de auditoría mediante bitácoras y diferentes comandos del Shell del sistema operativo o programa informático que se esté utilizando.

Tipos de Usuario

En GNU/Linux se manejan 3 tipos de usuarios: el usuario normal, el usuario especial y el usuario raíz (root). Estos se describen a continuación:

Usuario Normal: Los usuarios pertenecientes a este tipo son los de uso “normal” en un entorno de trabajo GNU/Linux, los usuarios que se utilizan para escuchar música, levantar un texto, navegar por internet o utilizar el Sistema hecho a la medida de una empresa. Cada uno de estos usuarios dispone de un directorio de trabajo, ubicado generalmente en /home. Tienen la libertad de personalizar su directorio, crear carpetas, guardar archivos, pero únicamente en este directorio, o bien en los directorios y archivos que les pertenezcan.



Tienen solo privilegios completos en su directorio de trabajo o HOME. En las distribuciones actuales de GNU/Linux se les asignan un número de UID superior a 1000.

Usuario Especial: También llamadas cuentas del sistema. Son usuarios que cumplen funciones específicas normalmente para brindar un servicio. Entre estos usuarios podemos encontrar bin, daemon, apache, squid, gdm. Estos se crean a la hora instalar el sistema operativo o algún programa que los requiera para su funcionamiento. Su UID se asigna entre 1 y 100 en los sistemas operativos GNU/Linux actuales.



Usuario Raíz (Root): También se le llama Super Usuario o Usuario Administrador. Es la cuenta de usuario con privilegios sobre todos los procesos y ficheros. Este usuario es el que ejecuta las tareas de mantenimiento y configuración del sistema, y la parte más delicada es que puede inclusive configurar el kernel y controladores. Lo más recomendable al autenticarse como usuario raíz es realizar las tareas respectivas y luego cerrar la sesión, esto para evitar tocar archivos importantes del sistema por equivocación, o bien para frenar el uso de usuarios no deseados.



Administración de Usuarios

La administración de procesos y servicios en UNIX – GNU/Linux se basa en archivos de configuración. Aunque la administración de usuarios no es cosa de todos los días, es muy importante conocer su funcionamiento pues son una parte imprescindible a la hora de ejecutar comandos o acceder a archivos y directorios. Entre los archivos de configuración para la administración de usuarios se pueden encontrar:

/etc/passwd:

Este archivo es el que encarga de almacenar los datos de los usuarios del sistema operativo. Se crea en la instalación del sistema operativo cuando se agrega el Usuario Root y los Usuarios Especiales. En el también se almacenan los usuarios normales. La configuración se realiza en una línea de código por usuario, la cual contiene 7 partes separadas por “:” y sin espacios:

```
metatron616:x:1001:1001:Gonzalo Guevara:/home/metatron616:/bin/bash
1      2      3      4      5      6      7
```

- *Parte1:* Nombre de Usuario para el inicio de Sesión, no se puede repetir en este archivo de configuración.
- *Parte2:* Contraseña, si se encuentra encriptada lleva el valor de "x" y sus valores se encuentran en el archivo shadow.
- *Parte3:* UID (número de identificación del usuario), es un valor único.
- *Parte4:* GID (número de identificación del grupo), número del grupo principal al cual el usuario pertenece.
- *Parte5:* Comentarios. Normalmente se utiliza para el nombre completo del usuario
- *Parte6:* Directorio de trabajo principal, normalmente se encuentra en “/home”
- *Parte7:* Shell que va a utilizar el usuario de manera predeterminada.

Para configurar varios usuarios de nuestro Sistema Operativo, debemos agregar tantas líneas como usuarios necesitemos. Por ejemplo, si quisiéramos agregar a Armando Rodríguez y a María Zamora a nuestro archivo de configuración se tendría que agregar una linea para cada uno quedando de la siguiente manera el archivo passwd:

```
metatron616:x:1001:1001:Gonzalo Guevara:/home/metatron616:/bin/bash
armando:x:1002:1002:Armando Rodriguez:/home/armando:/bin/bash
maria:x:1003:1003:María Zamora:/home/maria:/bin/bash
```

Este archivo normalmente no se edita “a mano”, para esto normalmente se utilizan comandos especiales, que se verán más adelante en esta misma sección del documento. Pero siempre es bueno saber su configuración para así poder hacer configuraciones más avanzadas.

/etc/shadow

El archivo “passwd” es muy útil, pero puede ser accesado por todos los usuarios del sistema para su lectura, Esto lo hace inseguro a la hora de guardar las contraseñas, Por ello se creó una extensión de este archivo con mayor seguridad, solamente el usuario raíz puede accederlo. Este archivo se llama “shadow” y se encarga de guardar, entre otras cosas, la encriptación de la contraseña de los usuarios del sistema que requieran una contraseña segura. Además guarda datos concernientes a la contraseña para establecer más seguridad. En total se divide en 9 partes:

```
root:$6$ZXBdYqrR$86NPeJP9xQYxUlzC3m2/YzKPfyYKp4.sw6Kt8vvSuFsvZA4T5S3gcwkD9eN5y4NSXg6PWG7c5/Z1D9fTxUONO.:15148:0:99999:7:-1::
1      2      3      4      5      6      7      8      9
```

- *Parte1:* Nombre de Usuario para el inicio de Sesión, no se puede repetir en este archivo de configuración.
- *Parte2:* Hash de cifrado de la contraseña del usuario.
- *Parte3:* Número de días que han pasado desde el 1 de enero de 1970¹ hasta la última fecha de modificación de la contraseña.
- *Parte4:* Número de días que deben transcurrir para que se pueda cambiar de nuevo la contraseña.

¹ Nota del autor: Los Epoch son fechas significativas en la historia, en el caso de los sistemas Unix y Unix-Like se toma en cuenta al 1 de enero de 1970, llamándolo “UNIX Epoch”. En el archivo “/etc/shadow” se hace una modificación de este, en lugar de tomar los segundos, la numeración se establece tomando en cuenta los días pasados a partir de esta fecha hasta una fecha específica. Para más información se puede consultar: http://es.wikipedia.org/wiki/Tiempo_Unix

- *Parte5:* Número de días para cambiar la contraseña. Con este dato se puede definir la fecha de expiración de la contraseña.
- *Parte6:* Número de días antes de la expiración de la contraseña en los que se notificará el usuario al iniciar sesión.
- *Parte7:* Número de días que pasan después de la expiración de la contraseña tras los cuales de desabilita si no se realiza el cambio.
- *Parte8:* Número de días que pasarán desde el 1 de enero de 1970 hasta el día de expiración de la cuenta.
- *Parte9:* Reservado.

En la Parte2 de la línea de configuración en el archivo “shadow”, si se tiene únicamente un “*” quiere decir que es un UsuarioEspecial.

```
...
daemon:*:15089:0:99999:7:::
```

/etc/group

En este fichero se guarda toda la información relacionada con los grupos de usuarios² a los cuales pertenecen los usuarios del sistema. Su forma de trabajo es muy parecida a la del archivo “/etc/passwd”, utilizándose una línea de configuración por cada grupo que se quiera establecer. La línea de configuración se divide en 4 partes:

```
root:x:0:root
1      2 3      4
```

- *Parte1:* Nombre del grupo. Valor único para este archivo de configuración.
- *Parte2:* Hash de cifrado de la contraseña del grupo, si tiene una “x” quiere decir que no tiene contraseña.
- *Parte3:* GID (número de identificación del grupo). Valor único para este archivo de configuración.
- *Parte4:* Usuarios que pertenecen al grupo, si son varios van separados por “,” y sin espacios.

En las versiones modernas de GNU/Linux al insertar un nuevo usuario su grupo se crea automáticamente tomando para el GID el UID, en otras palabras, si el UID del Usuario es 1000, el GID del grupo de ese usuario será igualmente 1000.

Como ejemplo corto del archivo de configuración “/etc/group” podemos ver:

```
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:metatron616,invitado,analista
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
metatron616:x:1001:metatron616
armando:x:1002:armando
maria:x:1003:maria
```

2 Un grupo de usuarios es una política utilizada en sistemas operativos y otros sistemas informáticos para referirse a más de 1 usuario en su configuración y privilegios.
Véase http://es.wikipedia.org/wiki/Grupo_de_usuarios

Comandos en GNU/Linux

Conceptos Básicos³

Anteriormente se mencionó que en Unix y GNU/Linux todo son ficheros. La utilización de los ficheros se hace mediante programas informáticos, como aplicaciones o utilidades del sistema operativo. Para la utilización de los programas informáticos se realizan instrucciones llamadas “comandos”, ya sea en una línea de comandos, emulador de terminal o con una instrucción programada en algún lenguaje de programación. Por otra parte un Shell es un programa que le da al usuario físico una interfaz para acceder a los programas y utilidades del sistema operativo, en otras palabras, el que se encarga de interpretar los comandos antes mencionados. En Ubuntu Server, el Shell predeterminado es “bash”, y es el que se utilizará en el resto del capítulo. Aunque no vamos a ahondar mucho en el tema del Shell es importante conocer las siguientes utilidades pues son de mucha ayuda en el mantenimiento de un Servidor Unix – GNU/Linux y es casi imprescindible su conocimiento:

Símbolos del Sistema (Command prompt)

Los símbolos del sistema son uno o más caracteres que indican, básicamente, el tipo de usuario que está utilizando la máquina. En UNIX – GNU/Linux solamente los usuarios normales y el súper-usuario pueden utilizar el sistema, pues los usuarios especiales los utilizan los servicios para poder ejecutarse y carecen de la opción de autenticación. Además puede indicar el nombre del usuario, el nombre de la máquina y la carpeta sobre la cual se está trabajando, como lo hace Ubuntu-Server.

Al inicio de la línea de comando nos aparecerá el nombre del usuario seguido del nombre del servidor con la siguiente nomenclatura:

```
nombre_usuario@nombre_maquina
```

Si nos encontramos en una carpeta diferente a la carpeta “/home/” de nuestro usuario, a la nomenclatura anterior se le preceden “:” y la ruta en la que nos encontramos, de la siguiente manera:

```
nombre_usuario@nombre_maquina:/ruta
```

Y por último tenemos el carácter que nos dice con qué tipo de usuario estamos trabajando. Cuando se está utilizando el SO como un usuario normal aparece un “\$” al inicio de la línea de comandos. En el terminal cuando iniciamos nuestro sistema operativo a modo consola, normalmente nos pregunta el “login”, donde escribimos el nombre de nuestro usuario, después de autenticarnos debería aparecer algo como:

```
nombre_usuario@nombre_maquina:/ruta$
```

Si queremos cambiar de usuario, el comando para poder hacerlo sería “su”, seguido del nombre del usuario. Por ejemplo:

```
$ su estudiante
```

Cuando se está utilizando el SO como usuario raíz aparece un “#” al inicio de la línea de comandos.

```
root@nombre_maquina:/ruta#
```

Si se desea autenticar como usuario raíz debe escribir el comando “sudo su”⁴:

```
$ sudo su
```

En el resto del documento, para las instrucciones y comandos que se van a presentar, como aclaraciones o ejemplos, únicamente se dará el carácter que diferencia al tipo de usuario (“\$” o “#” según su efecto), omitiendo el resto de los símbolos del sistema.

Concatenación de comandos

Bash nos permite concatenar comandos utilizando el operador lógico AND (“&&”) además de los paréntesis redondos (“(“ y ””). Su utilización es muy parecida a la que se le daría en la matemática a estos operadores.

Empecemos con el uso del AND. Este nos sirve para ejecutar varios comandos en una sola línea:

```
$ parámetro1 && parámetro2 && parámetroN
```

³ Nota del autor: Vale la pena repasar los conceptos básicos una vez terminado de leer el documento.
⁴ Nota del autor: En Ubuntu y cualquiera de sus derivados se puede utilizar el comando “sudo parámetro” para ejecutar cosas como súper-usuario sin tener que autenticarse normalmente como tal. Pero en lo personal recomiendo no utilizar esto, sino autenticarse como normalmente como usuario raíz y luego de ejecutar lo que es menester cerrando seguido dicha sesión y seguir trabajando como un usuario normal.

Por otro lado los paréntesis, al igual en matemática, nos sirven para unir expresiones en una sola orden y que se ejecuten juntas, por así decirlo:

```
$ (parámetro1 && parámetro2)
```

Esto último es útil para un sin fin de situaciones, en el transcurso del documento veremos varias de sus implementaciones.

Redireccionamiento

En UNIX – GNU/Linux la interacción con el usuario se realiza mediante 3 canales: salida estándar (stdout), entrada estándar (stdinn) y error estándar (stderr). Estos canales se pueden interconectar con archivos, dispositivos y con canales de otros programas. Antes que se ejecute un comando, su entrada y/o salida puede ser redirigida utilizando alguna notación especial interpretada por el Shell.

Redirección de salida estándar: Abre el archivo “nombre_archivo” y escribe los datos guardados en un “parámetro”, el cual puede ser el resultado de un comando o una variable. Si el archivo no existe se crea, y si existe le da valor vacío y empieza a escribir en él.

```
$ parámetro >[|]nombre_archivo
```

Esto nos sirve cuando queremos documentar el resultado de un comando, crear bitácoras de errores, o bien bitácoras de acceso, etc. Para ilustrar más el concepto podemos realizar el siguiente comando:

```
$ ls /home/estudiante/ > /home/estudiante/archivo.prueba
```

La instrucción “ls” nos permite ver el contenido de una carpeta. El comando anteriormente ejecutado nos guardará una lista con los archivos y carpetas que se encuentran dentro de la carpeta “/home/estudiante/” en el archivo llamado “ /home/estudiante/archivo.prueba”.

Redirección del error estándar: Abre el archivo “nombre_archivo” y escribe los errores generados a la hora de ejecutar un “parámetro”, el cual puede ser el resultado de un comando o una variable

```
$ parámetro 2> nombre_archivo
```

Esto nos permite por ejemplo crear una bitácora de errores a la hora de ejecutar un comando, por ejemplo:

```
$ cat archivo.txt 2> err.txt
```

La instrucción “cat” a groso modo nos muestra el contenido de un archivo, si este archivo no existe nos devolverá un error como:

```
cat: archivo.txt: No existe el fichero o el directorio
```

Este error quedará escrito en el archivo “err.txt”.

Redirección de entrada: Toma los datos de un archivo “nombre_archivo” abriéndolo solo en modo lectura y pasar su contenido a una instrucción “parámetro”.

```
$ parámetro < nombre_archivo
```

Esto es útil cuando el contenido de un archivo nos sirve para ejecutar un parámetro repetidamente. Por ejemplo al enviar un mensaje a varias cuentas de correo electrónico con el contenido de un “archivo.txt”

```
$ mail pepe@correo.com estudiante@correo.com < archivo.txt
```

Adjuntar a una salida redirigida: Abre un archivo “nombre_archivo” y agrega al final de este una nueva línea con el contenido del “parámetro” dado. Si el archivo no existe crea uno nuevo con contenido vacío y luego escribe lo solicitado en él.

```
$ parámetro >> nombre_archivo
```

Esto nos puede servir para hacer una bitácora en el “archivo.prueba” con el contenido de archivos en la carpeta “/home/estudiante” en una fecha dada y guardando así los datos anteriores:


```
$ (date && ls /home/estudiante/) >> archivo.prueba
```

Si ejecutamos varias veces el mismo comando, los datos anteriores seguirán en el y se agregarán cada vez más datos nuevos, haciendo una bitácora sencilla con las carpetas que puede servir para una futura auditoría.

Tuberías

Es la instrucción encargada de unir la salida estándar de una instrucción con la entrada estándar de otra instrucción. En otras palabras, une la entrada de la “instrucción1” con la salida de la “instrucción2”. Esta unión se realiza mediante el carácter “|”.

```
$ instrucción1 | instrucción2
```

Para entender mejor el concepto veamos el siguiente ejemplo:

```
$ ls /home | grep "estudiante"
```

En este comando la instrucción “*ls /home*” muestra la lista de archivos y directorios que se encuentran en la carpeta /home. La instrucción “*grep*” por otro lado busca una palabra determinada, “*estudiante*” para este caso, y muestra las líneas que coincidan con ella. Al utilizar la tubería, se está buscando todos los archivos y carpetas que tengan el nombre “*estudiante*” en el directorio /home.

Las tuberías son de suma utilidad, y su uso es muy variado dependiendo de las instrucciones y/o comandos que se convienen.

Utilidades Para el manejo de ficheros y directorios.

ls: Lista información sobre ficheros presentes en un directorio. Su sintaxis es “ls [opción] [archivo/directorio]”. Su utilización sencilla sería:

```
# ls /home/estudiante
```

Mostrando únicamente los archivos presentes en el directorio “/home/estudiante”. Si se desea conocer las propiedades de dichos archivos se puede recurrir a la opción “-la” que despliega todas las propiedades y los archivos en una lista larga:

```
# ls -la /home/estudiante
```

mv: Mueve un fichero o directorio de su ubicación actual, a una nueva. En otras palabras, realiza una copia del archivo en otra dirección y elimina el archivo original. Su sintaxis sería “mv *_fichero_a_copiar _rutanueva*”. Para su utilización se ejecuta:

```
# mv /_ruta_vieja/_archivo /_ruta_nueva
```

Si se quiere cambiar de nombre al archivo cuando se mueve:

```
# mv /_ruta_vieja/_Nombre_Archivo_Viejo /_ruta_nueva/_Nombre_Archivo_Nuevo
```

cp: Copia un archivo o directorio, manteniendo el archivo original. Si se desea copiar un archivo, se hace de la siguiente manera:

```
# cp /_ruta_origen/_archivo_origen /_ruta_destino/
```

Si se quiere copiar un archivo con otro nombre:

```
# cp /_ruta_origen/_archivo_origen /_ruta_destino/_nuevo_nombre
```

Si se quisiera copiar una carpeta es el mismo procedimiento, pero hay que agregar la opción “-R” que establece la recursividad:

```
# cp -R /_directorio_a_copiar/ /_ruta_destino/
```

chmod: Establece o modifica los permisos (modos) de un archivo o directorio. En el se definen los permisos para el dueño del archivo/directorio, el grupo, y los otros que puedan accederlo. Este comando puede trabajar mediante 2 maneras:

Modo Octal:

Este sistema se basa en 3 dígitos, cada uno a base8, representando el primero el dueño, el segundo el grupo y el tercero los demás usuarios. Su sintaxis es sintaxis “*chmod ### [archivo]*”. El significado del valor de cada dígito viene expresado en la siguiente tabla:

Número	Binario	Lectura (r)	Escritura (w)	Ejecución (x)
0	000	✗	✗	✗
1	001	✗	✗	✓
2	010	✗	✓	✗
3	011	✗	✓	✓
4	100	✓	✗	✗
5	101	✓	✗	✓
6	110	✓	✓	✗
7	111	✓	✓	✓

Fuente: <http://es.wikipedia.org/wiki/Chmod>

Para su mayor comprensión se explica en el siguiente cuadro:

```
chmod 766 file.txt      # brinda acceso total al dueño
                        # y lectura y escritura a los demás
chmod 770 file.txt      # brinda acceso total al dueño y al grupo
                        # y elimina todos los permisos a los demás usuarios
chmod 635 file.txt      # Permite lectura y escritura al dueño,
                        # escritura y ejecución al grupo,
                        # y lectura y ejecución al resto
```

Fuente: <http://es.wikipedia.org/wiki/Chmod>

Modo carácter:

Posee 3 modificadores que permiten realizar la tarea:

- + – añade un modo
- – – elimina un modo
- = – especifica un modo (sobrescribiendo el modo anterior)

Por ejemplo:

```
chmod +r arch.txt      # agrega permisos de lectura a todos los
                        # usuarios
chmod u+w arch.txt      # agrega permisos de escritura al dueño
chmod -x arch.txt      # elimina el permiso de ejecución a todos
                        # los usuarios
chmod u=rw,go= arch.txt # establece los permisos de lectura y escritura
                        # al dueño y elimina todos los permisos a
                        # los demás usuarios
```

Fuente: <http://es.wikipedia.org/wiki/Chmod>

rm: Elimina un archivo o directorio. Para eliminar un archivo:

```
$ rm /ruta/fichero
```

Para eliminar un directorio se le debe agregar la opción “-R” para establecer la recursividad:

```
$ rm -R /ruta
```

mkdir: Crea un nuevo directorio.

```
$ mkdir /ruta/directorio_nuevo
```

cat: Muestra el contenido de un archivo.

```
$ cat archivo.prueba
```

grep: Busca una cadena de texto. Esto lo hace en un directorio, fichero o entrada estandar. Es de mucha utilidad utilizarlo mediante una tubería.

En el siguiente ejemplo, mostrará todas las líneas que contengan la palabra “arbolito” en el archivo “archivo.prueba”.

```
$ cat archivo.prueba | grep "arbolito"
```

Este otro ejemplo mostrará todos los archivo con terminación “.txt” que se encuentren en el directorio “/home/estudiante”:

```
$ ls /home/estudiante | grep *.txt
```

less: Es un paginador que permite retroceso y avance en un documento. Este muestra las primeras 25 líneas del documento y da la opción para desplazarse en él mediante las flechas del teclado. Es de mucha utilidad utilizarlo mediante una tubería.

```
$ cat archivo.prueba | less
```

nano: Este es un editor de texto en consola, su uso es muy intuitivo, el cursor se maneja con las flechas del teclado, entre otras bondades. Puede modificar un archivo existente o crear uno nuevo en caso de que el nombre de archivo establecido no exista. Para su uso se ejecuta:

```
$ nano /ruta/archivo.prueba
```

Comandos de Administración de Usuarios:

useradd:

Este comando crea un nuevo usuario, para ejecutarse tiene que hacerse como usuario raíz. Su sintaxis es “*useradd [options] [LOGIN]*”. Por ejemplo:

```
# useradd -D metatron616
```

Entre las opciones más comunes para su utilización se encuentran:

Opción		Descripción
-D	--defaults	Otorga los parámetros predeterminados a todas las opciones posibles de useradd.
-c	--comment	Establece el campo de comentarios, utilizado normalmente para el nombre completo del usuario (parte 5 de la línea de configuración de “/etc/passwd”)
-m	--create-home	Crea el directorio de usuario si este no había sido creado con anterioridad
-p	--password	Establece la contraseña del usuario
-g	--gid	Establece el número de identificación o nombre del grupo principal al cual pertenecerá el nuevo usuario
-G	--groups	Establece los grupos secundarios a los cuales pertenecerá el usuario. Si se quieren agregar varios grupos se deben separar por “,” y sin espacios.
-u	--uid	Establece el número de identificación del usuario. Este debe ser un número entero mayor a 0. Normalmente los números entre 0 y 999 están reservados para cuentas especiales del sistema.
-U	--use-group	Crea un grupo con el mismo nombre del usuario y se incluye como grupo principal del usuario creado.

Para tener un ejemplo de inserción de usuario con varias opciones podemos ver la siguiente línea de comando:

```
# useradd -c 'Gonzalo Guevara' -d '/home/metatron616' -m -U -G admin -p XxXxXxXx metatron616
```

usermod:

El comando modifica un usuario existente. Para poder ejecutar este comando hay que hacerlo como usuario raíz, siendo su sintaxis “*usermod[options] [LOGIN]*”. Por Ejemplo:

```
# usermod -l gonzalo metatron616
```

Entre las opciones más comunes para su utilización se encuentran:

<i>Opción</i>		<i>Descripción</i>
-c	--comment	Modifica el campo de comentarios, utilizado normalmente para el nombre completo del usuario (parte 5 de la línea de configuración de “ <i>/etc/passwd</i> ”)
-l	--login	Cambia el LOGIN del usuario por un NUEVO_LOGIN.
-d	--home	Modifica el directorio principal del usuario, donde el usuario tiene todos los permisos de escritura y lectura.
-m	--move-home	Mueve el contenido del directoro “home” a una nueva localización. Esta opción solo se puede utilizar en convinación de “-d”
-p	--password	Establece la contraseña del usuario
-g	--gid	Modifica el número de identificación o nombre del grupo principal al cual pertenecerá el nuevo usuario
-G	--groups	Modifica los grupos secundarios a los cuales pertenecerá el usuario. Si se quieren agregar varios grupos se deben separar por “,” y sin espacios. Si el usuario era miembro de un grupo no listado en la opción, el usuario será remodivo de dicho grupo.
-u	--uid	Modifica el número de identificación del usuario. Este debe ser un número entero mayor a 0. Normalmente los números entre 0 y 999 están reservados para cuentas especiales del sistema.

Si se quisiera utilizar varias opciones para modificar el usuario se puede ver:

```
# usermod -l gonzalo -d /home/gonzalo -m -P XxxXXxXXx metatron616
```

userdel:

El comando elimina una cuenta de usuario existente y sus archivos relacionados. Para poder ejecutar este comando hay que hacerlo como usuario raíz, siendo su sintaxis “*userdel [options] [LOGIN]*”. Por Ejemplo:

```
# userdel metatron616
```

Entre las opciones más comunes para su utilización se encuentran:

<i>Opción</i>		<i>Descripción</i>
-f	--force	Fuerza la eliminación de la cuenta de usuario aunque esta se encuentre autenticada en ese momento.
-r	--remove	Los archivos del usuario a eliminar en el directorio “/home” también serán eliminados.
-h	--help	Muestra el mensaje de ayuda y se cierra

passwd:

Cambia la contraseña de una cuenta de usuario. Un usuario normal solo podría cambiar su contraseña. Pero el usuario raíz puede cambiar la contraseña de cualquier cuenta con este comando, además de cambiar varias de sus opciones de validez. El usuario normal solo podría ejecutar la sintaxis “passwd”.

```
$ passwd
```

A lo cual preceden las siguientes instrucciones:

```
Cambiando la contraseña de metatron616.

(actual) contraseña de UNIX:
Introduzca la nueva contraseña de UNIX:

Vuelva a escribir la nueva contraseña de UNIX:
```

En el caso de ejecutar el comando con el usuario raíz la sintaxis sería “passwd [opcion] LOGIN”. Por ejemplo:

```
$ passwd estudiante
```

Entre las opciones más comunes para su utilización se encuentran:

<i>Opción</i>		<i>Descripción</i>
-S	--status	Muestra información sobre el estado de la contraseña en el usuario
-a	--all	Solo s e puede utilizar con -S, y muestra el estado de las contraseñas de todos los usuarios.
-e	--expire	Expira inmediatamente la contraseña de la cuenta de usuario. El usuario tendrá que cambiar de contraseña la próxima vez que inicie sesión.
-d	--delete	Hace que el campo de contraseña se encuentre vacío, es una forma rápida para desabilitar la contraseña en la cuenta.
-n	--mindays	Establece el número mínimo de días entre cambio de contraseña.
-x	--maxdays	Establece el número máximo de días entre cambio de contraseña.
-w	--warndays	Establecer el número de días de advertencia antes de un cambio de contraseña.

Para más información de cada comando se puede consultar la documentación desde el emulador de terminal, mediante el programa “man”, mediante el comando “man nombre_comando”. Por ejemplo:

```
$ man useradd
```

Administración de aplicaciones

apt-get:

La instrucción “*apt-get*” es la herramienta para la gestión de paquetes desde la línea de comandos. Se considera como como el sistema de bajo nivel de otras herramientas de gestión de paquetes. Está ligado a las fuentes presentes en el archivo de configuración “*/etc/apt/source.list*”, si las fuentes proceden del internet se debe tener conexión a ella para poder realizar las tareas. A estas fuentes o lugares de donde se obtienen los paquetes se les llama repositorios. Normalmente puede ser utilizado solo por el usuario raíz. Su utilización es muy sencilla y se puede dividir en 4 acciones principales:

Actualización de los repositorios: Actualiza la lista de paquetes disponibles y sus versiones en los repositorios que se encuentran en el archivo “*/etc/apt/source.list*”. Es muy importante actualizar estos repositorios periódicamente, pues las nuevas versiones de los paquetes contienen la corrección de errores que pueden afectar el buen funcionamiento o la seguridad de estos mismos paquetes y todo nuestro sistema operativo. Para realizar la actualización de los repositorios se ejecuta la siguiente orden:

```
# apt-get update
```

Actualización completa de los paquetes: Una vez actualizados los repositorios se procede a actualizar los paquetes. Esto se hace mediante el comando:

```
# apt-get upgrade
```

Instalación de paquetes: Se pueden instalar paquetes que se encuentren en los repositorios. La instalación de paquetes mediante apt es muy sencilla, pues busca todas las dependencias necesarias. Esto se ejecuta mediante el siguiente comando:

```
# apt-get install paquetel paquete2 paqueteN
```

Como se ve, apt tiene la posibilidad de instalar N paquetes en la misma línea de comando. Para poder instalar un paquete se debe conocer el nombre de este. Un ejemplo sencillo de su utilización sería:

```
# apt-get install nmap
```

Con esto instalamos “nmap”, el cual es un programa de monitorización de red.

Eliminación de paquetes: En algunos casos es necesario eliminar un programa, ya sea porque se encontró un remplazo o simplemente porque ya no es de utilidad. Una eliminación básica, pero que deja los archivos de configuración del programa, sería:

```
# apt-get remove nombre_paquete
```

Si deseamos, por ejemplo, eliminar el paquete “nmap” instalado anteriormente se procedería:

```
# apt-get remove nmap
```

Ahora, si se desea eliminar por completo el paquete, sin dejar rastro alguno, se procede a ejecutar:

```
# apt-get purge nombre_paquete
```

Utilizando de nuevo el ejemplo de eliminación, pero ahora por completo, del programa nmap:

```
# apt-get purge nmap
```

.

dpkg

A pesar que “apt-get” es de gran ayuda y ofrece muchas opciones automatizadas como la búsqueda de paquetes y sus dependencias en repositorios ya establecidos, muchas veces es necesario instalar programas manualmente pues no se encuentran en los repositorios y solo se tiene el archivo “.deb” para su instalación. En este escenario es donde entra “dpkg”, encargado de instalar paquetes, balga el redundancia, empaquetados en archivos “.deb”. Su sintaxis es “dpkg [opcion] accion”

Entre las opciones más comunes para su utilización se encuentran:

<i>Opción</i>		<i>Descripción</i>
-i	--install	Instala un paquete seleccionado.
-r	--remove	Elimina un paquete instalado dejando sus ficheros de configuración.
-P	--purge	Elimina un paquete instalado con sus ficheros de configuración.

Para instalar un paquete:

```
# dpkg -i nmap.deb
```

Si se tiene en una carpeta, donde está el paquete que se quiere instalar y sus dependencias, todos los paquetes deben estar en la misma carpeta y se debe de estar dentro de ella, entonces se ejecuta:

```
# dpkg -i *.deb
```


Comandos útiles:

ac	Imprime estadísticas acerca del tiempo que han estado conectado los usuarios.
adduser	Ver useradd.
alias	Crea atajos de comandos, lista los alias actuales.
apt-get	Herramienta de actualización/instalación remota de paquetes en sistemas basados en debian.
arp	Permite obtener/manipular la lista de direcciones MAC/Ip que el sistema ve.
arping	Envía ARP REQUEST a otros equipos en la red.
arptables	Firewall similar en funciones a iptables pero para control de tráfico de protocolo arp.
at	Programa trabajos, comandos, scripts para su ejecución posterior.
atq	Lista los trabajos programados pendientes de ejecutar por el comando at.
awk	Análisis y procesamiento de patrones en archivos y listados.
basename	Permite eliminar la ruta del nombre de un archivo.
bc	Calculadora y lenguaje matemático, muy potente.
biosdecode	Información sobre el BIOS.
blkid	Muestra atributos de dispositivos de bloque (discos, usb, etc.) tales como LABEL y UUID, entre otros.
bzcat	Descomprime archivos comprimidos o empaquetados mediante bzip2.
bzip2	Compresor / descompresor de archivos.
bzmore	Permite ver el contenido de archivos comprimidos o empaquetados mediante bzip2.
cal	Despliega un calendario.
cat	Muestra el contenido de archivos y concatena archivos.
cd	Cambiar de directorio.
cfdisk	Herramienta de particionamiento de discos, usada en sistemas debian principalmente.
chage	Permite cambiar la información (expiración, caducidad, etc) de la contraseña de un usuario.
chattr	Cambia atributos extendidos de archivos y directorios
chfn	Cambia la información usada en finger.
chgrp	Cambia el grupo de un archivo(s) o carpetas(s).
chkconfig	Controla/consulta el modo en que los servicios se ejecutan o no al inicio del sistema.
chmod	Cambia los permisos de un archivo(s) o carpetas(s).
chown	Cambia el propietario de un archivo(s) o carpetas(s).
chpasswd	Actualiza passwords o contraseñas en modo batch. Puede actualizar contraseñas de grupos de usuarios.
chroot	Ejecuta comandos de root en un shell restringido a un directorio y sus subdirectorios.
chsh	Cambia tu shell por defecto o shell de login.
cleanlinks	Limpia enlaces simbólicos que no tengan relación y también remueve directorios vacios.
clear	Limpia la terminal.
cmp	Compara dos archivos byte por byte.
convertquota	Convierte de los viejos formatos quota.user y quota.group a los nuevos formatos de aquota.user y aquota.group.
cpio	Copia, crea, comprime y extrae archivos en distintos formatos y entre equipos o localmente.
crontab	Administra archivos cron para los usuarios y root.
curl	Permite descargar o transferir url's.
cut	Remueve secciones (columnas principalmente) de cada línea de un archivo o archivos.
date	Muestra/establece la fecha y hora actual.
dc	Calculadora interactiva.
dd	Convierte y copia archivos y sistemas de archivos.
ddate	Muestra la fecha en formato del calendario Discordante.
df	Muestra el uso de espacio de discos duros o particiones.
diff	Busca y muestra diferencias entre archivos.
dig	Utilería para consultas a servidores DNS.
dircolors	Configuración de colores para el comando ls.
dirs	Permite mostrar, manipular la lista de directorios utilizados en la pila. (ver popd y pushd)
dmesg	Muestra los mensajes del arranque del sistema (boot).
dmidecode	Lista hardware del equipo directamente del BIOS. (también: lshw)
dos2unix	Convierte archivos de formato MS-DOS a formato Unix/Linux.
du	Muestra el uso de espacio de archivos y directorios.
dump	Permite la creación de respaldos para los sistemas de archivos ext2 y ext3.
echo	Imprime una línea de texto, variables, o contenido a un archivo.
edquota	Administra el control de cuotas de disco de usuario y grupos.
egrep	Es igual que el comando 'grep -E', para uso de expresiones regulares.
eject	Desmonta y expulsa un medio removible, como cdroms.
env	Ejecuta un programa en un entorno modificado.
ethtool	Permite desplegar o cambiar valores de una tarjeta de red.
exit	Sale del shell o terminal actual.
expect	Permite crear secuencias de diálogos y programar sesiones interactivas con otros comandos o scripts.
export	Exporta el valor de una variable.

exportfs	Mantiene una lista de sistemas de archivos del tipo NFS que han sido exportados.
expr	Evaluador de expresiones matemáticas.
factor	Encuentra los números primos de un número dado.
fc	Lista, edita y reejecuta comandos previamente ejecutados.
fdisk	Herramienta para particionar discos, común a casi todas las distros.
fgrep	Es igual que 'grep -F' para uso de expresiones regulares en búsquedas de archivos y listados.
file	Determina el tipo de archivo.
find	Búsqueda de archivos, multitud de opciones de búsqueda.
findfs	Busca un sistema de archivos por UUID o LABEL (etiqueta).
findsmb	Lista información sobre equipos que respondan a paquetes SMB. Lista una red Windows. (Parte del paquete Samba)
finger	Muestra información sobre los usuarios del sistema.
fortune	Imprime un adagio al azar.
fping	Permite mandar paquetes ICMP (pings) a múltiples equipos en una red y determinar si están vivos o no.
free	Muestra el espacio usado y libre de memoria RAM y Swap.
fsck	Herramienta para verificar/reparar sistemas de archivos.
fuser	Identifica procesos utilizando archivos o conexiones (sockets).
gawk	Análisis y procesamiento de patrones en archivos y listados. (versión gnu)
gcc	Compilador de C y de C++ de GNU.
gedit	Editor de textos de gnome.
gpasswd	Permite la administración del archivo /etc/group
gpg	Herramienta de encriptación y de generación de certificados de seguridad (opengpg).
grep	Busca patrones de cadenas dentro de archivos.
groupadd	Crea un nuevo grupo en el sistema.
groupdel	Elimina un grupo en el sistema.
groupmod	Modifica un grupo en el sistema.
groups	Imprime los grupos a los que pertenece un usuario.
gzip	Comprime/expande archivos.
halt	Apaga el equipo.
hdparm	Establece y muestra características sobre los discos duros.
head	Despliega las primera líneas de un archivo.
help	Ayuda sobre los comandos internos de bash.
history	Muestra el historial de comandos del usuario.
host	Utileria de consulta a servidores DNS.
hostname	Despliega el nombre del equipo.
htpasswd	Administra archivos de usuario/contraseña para autenticación básica de Apache.
hwclock	Muestra/Establece la fecha/hora del bios o hardware. (Fecha/Hora del sistema con date)
id	Muestra el UID (User ID) y GID (Group ID) del usuario
ifconfig	Muestra/Configura las interfaces de red del sistema.
ifstat	Pequeña utileria que permite observar estadísticas de las interfaces de red en tiempo real.
init	Control de inicialización de un nivel de ejecución.
insmod	Inserta módulos en el kernel.
ipcalc	Realiza cálculos simples sobre direcciones IP.
ipcount	Identificación de rangos de red, cálculo de IP's.
iptab	Muestra una tabla de direcciones IP de acuerdo al prefijo CIDR
iptables	Herramienta de configuración del firewall de Linux.
iptraf	Analizador de tráfico de red en modo de texto.
iwconfig	Configura una tarjeta de red inalámbrica.
iwlist	Obtiene información detallada de una tarjeta inalámbrica.
jobs	Muestra los trabajos del usuario en suspensión o en background.
kate	Editor de textos de KDE.
kill	Termina procesos, mas correctamente envía señales a procesos.
killall	Termina procesos del mismo nombre o conjunto.
last	Muestra información de los últimos usuarios logueados.
lastb	Muestra información de los últimos intentos fallidos de loguearse.
less	Muestra el contenido de un archivo, permite búsquedas y movimiento hacia atrás y adelante.
ln	Crea enlaces (accesos directos) suaves y duros de archivos y directorios.
locale	Información específica sobre las variables de entorno locales.
locate	Indexa y busca archivos. Mas seguro utilizar slocate.
losetup	Define y controla dispositivos del tipo 'loop'.
lpq	Muestra los documentos para imprimir en la cola de impresión.
lpr	Añade un documento a la cola de impresión.
ls	Lista archivos y directorios.
lshw	Lista hardware del equipo directamente del BIOS. (también: dmidecode)

lsmod	Muestra el estatus de los módulos en el kernel.
lsuf	Muestra archivos abiertos de un programa en ejecución, o de un usuario, proceso, etc.
lspci	Lista los dispositivos pci del sistema.
lsusb	Lista los dispositivos usb del sistema.
mail	Envía y recibe correos.
man	Muestra el manual del comando indicado.
mc	Manejador de archivos con soporte de mouse en modo de texto, no todas las distro lo tienen.
mcedit	Editor de textos de mc.
md5sum	Comprueba (y genera) archivos con firma de certificación md5.
mkdir	Crea directorios.
mkfs	Construye un sistema de archivos de Linux.
mkpasswd	Generador de contraseñas. (Paquete del programa 'expect').
modinfo	Muestra información acerca de los módulos del kernel.
modprobe	Herramienta que añade/remueve módulos del kernel.
more	Paginador similar a less pero menos funcional, ya que sale avanza y no retrocede.
mount	Monta dispositivos de almacenamiento en particiones indicadas.
mtools	Conjunto de utilidades para acceder a discos DOS desde Linux.
mv	Mueve archivos y directorios.
netstat	Herramienta de red que muestra conexiones, tablas de ruteo, estadísticas de interfaces, etc.
nice	Ejecuta un programa con una prioridad de ejecución distinta a la normal.
nohup	Ejecuta un programa inmune a los hangups y sin salida a una terminal.
openssl	Control, administración, generación de certificados de seguridad.
partprobe	Indica al sistema operativo de los cambios indicados en /etc/fstab
passwd	Cambia la contraseña del usuario indicado.
ping	Manda un echo_request (solicitud de eco) a un equipo en al red.
pkill	Manda señales a procesos basado en sus atributos.
popd	Remueve entradas (directorios utilizados) de la lista de directorios utilizados en la pila. (ver dirs y pushd)
pr	Formatea o convierte archivos de texto para imprimirlos.
ps	Muestra los procesos del sistema o del usuario o ambos.
pstree	Muestra los procesos en forma de árbol.
pushd	Agrega entradas (directorios utilizados) en la lista de directorios (pila o stack). (ver dirs y popd)
pwck	Verifica la integridad del archivo /etc/passwd
pwconv	Agrega o establece la protección shadow el archivo /etc/passwd.
quota	Permite ver el uso de cuotas por usuario.
quotacheck	Crea, verifica, administra sistemas de cuotas de disco
quotaoff	Desactiva el control de cuotas de discos.
quotaon	Activa el control de cuotas de discos para usuarios y grupos.
rdesktop	Abre terminales gráficas hacia equipos Windows.
reboot	Reinicia el equipo.
renice	Cambia la prioridad de un proceso o programa en ejecución.
repquota	Reporte de uso de cuotas de disco.
resolveip	Resuelve la ip del dominio o host que se indique.
rev	Invierte las líneas de un archivo.
rm	Borra o elimina archivos.
route	Muestra/altera la tabla de ruteo IP.
rpm	Programa para la instalación/actualización/eliminación de paquetes, distros basadas en redhat.
runlevel	Muestra el nivel de ejecución actual y anterior del sistema.
scp	Copia archivos entre equipos, parte del paquete openssh (protocolo de comunicación encriptado).
screen	Administrador de terminales virtuales.
sed	Editor en línea que filtra y transforma archivos.
service	Ejecuta/detiene servicios en modo manual.
set	Muestra o establece el entorno de variables para el usuario actual.
sha1sum	Comprueba (y genera) archivos con firma de certificación sha1.
shopt	Habilita o deshabilita variables opcionales del comportamiento del shell.
shred	Elimina archivos de manera segura e irrecuperable.
shutdown	Apaga o reinicia el equipo.
sort	Ordena líneas de archivos y listas
ss	Utileria similar a netstat pero más básica, listados rápidos de sockets establecidos.
ssh	Programa de login remoto seguro, programa del paquete openssh (protocolo de comunicación encriptado).
startx	Inicia una sesión X.
su	Cambia del usuario actual al indicado.
sudo	Permite indicar que usuario ejecuta que comandos de root.
sync	Forza bloques en memoria a discos, actualiza el superbloque.
tac	Igual que cat, muestra y/o concatena archivos pero al revés.

tail	Muestra la parte final de un archivo.
tailf	Sinónimo del comando tail -f, permite ver en tiempo real la parte final de un archivo, es decir, conforme se va escribiendo, útil para monitorear bitácoras.
tar	Herramienta empaquetadora/compresora de archivos.
testparm	Revisa archivos smb.conf de samba por errores o correcciones.
time	Devuelve el tiempo en que se ejecutó el comando o programa indicado.
top	Muestra los procesos del sistema de manera interactiva y continua.
touch	Crea archivos vacios, cambia fechas de acceso y/o modificación de archivos.
tput	Cambia valores o capacidades de la terminal, en base a terminfo.
traceroute	Imprime la ruta de los paquetes de red hasta el destino indicado.
tty	Imprime el nombre de la terminal en la que se esta.
tzselect	Permite establecer una zona o huso horario.
umask	Establece una máscara de permisos para cuando se crean directorios y archivos.
umount	Desmonta sistemas de archivos.
unalias	Elimina alias de comandos, creados con el comando alias.
uname	Despliega información del sistema.
uniq	Omite o reporta sobre líneas repetidas en un archivo o listado.
units	Convertidor de unidades de un sistema a otro, soporta decenas de sistemas de medición.
up2date	Herramienta de actualización/instalación remota de paquetes, (usada en redhat, centos).
uptime	Muestra que tanto tiempo lleva prendido el equipo.
urpme	Programa del paquete urpmi para desinstalar o eliminar paquetes.
urpmi	Herramienta de actualización/instalación remota de paquetes, distros basadas en rpm (usada en mandriva).
useradd	Añade usuarios.
userdel	Elimina usuarios.
usermod	Modifica información de usuarios.
users	Muestra los nombres de usuario de todos los usuarios conectados actualmente al sistema.
vi	Editor visual de pantalla, editor de textos, que encuentras en todas las distros Linux.
vim	Igual que el vi pero mejorado.
visudo	Editor para el archivo de configuración /etc/sudoers de sudo.
vmstat	Proporciona información sobre la memoria virtual.
w	Muestra quien esta conectado al sistema y que esta haciendo.
wall	Manda un mensaje a todas las terminales.
warnquota	Configura /etc/warnquota.conf como complemento de mensajes para cuotas de disco.
wc	Cuenta palabras, líneas, caracteres de un archivo o listado.
wget	Descargador de archivos desde Internet, no interactivo.
whatis	Descripción corta, en una línea de un comando o programa.
whereis	Localiza el binario, fuentes y/o librerías, y documentación de un comado.
which	Muestra la ruta completa de un comando.
who	Muestra quien esta conectado al sistema.
whoami	Muestra el usuario actual.
xhost	Control de acceso para sesiones X.
xkill	Mata o termina a un cliente X, es decir, a un programa gráfico.
yes	Imprime una cadena repetidamente hasta que sea terminado o matado el comando.
yum	Herramienta de actualización/instalación remota de paquetes, distros basadas en rpm (usada en fedora, redhat y derivados).
zcat	Descomprime / muestra archivos comprimidos con gunzip (es idéntico a utilizar gunzip -c)
zenity	Despliega varios tipos de diálogos en X desde una terminal.
zless	Permite mostrar el contenido de archivos comprimidos.
zmore	Permite mostrar el contenido de archivos comprimidos.

Fuente: http://www.linuxtotal.com.mx/index.php?cont=info_admon_002 . Por Sergio González Durán

Referencias

http://wiki.ehas.org/index.php?title=Manual_de_l%C3%ADnea_de_comandos_linux

<http://es.kioskea.net/contents/linux/>

<http://www.cyberciti.biz/faq/understanding-etcgroup-file/>

<http://www.gnu.org/s/bash/manual/bash.html#Simple-Commands>

http://www.linuxtotal.com.mx/index.php?cont=info_admon_002

Agradecimientos:

Gracias a Sergio González Durán, del sitio web linuxtotal.com.mx, por toda la información brindada en sus artículos y su espíritu de colaboración. Y a tantos otros amigos que contribuyen con la comunidad compartiendo su conocimiento.