



Peruvian Linux Users Group (PLUG)

LINUX PARA NOVATOS

(Basado en Red Hat 6.1)

Escrito por : [Fernando Fauché Gonzalez](#)

CAPITULO I

UN BUEN PUNTO DE PARTIDA

CAPITULO I

UN BUEN PUNTO DE PARTIDA

Para el usuario acostumbrado a sistemas operativos como IBM OS/2, Microsoft DOS o Microsoft Windows, lo primero que debe comprender es que las diferencias entre ambos sistemas operativos pueden ser considerables ya que la filosofía del manejo de cada uno es totalmente diferente, de principio a fin.

Empezaremos citando las diferencias más relevantes para comenzar sin tropiezos.

El símbolo de “raíz”

A diferencia de los sistemas basados en IBM OS2 y Microsoft DOS, donde el caracter que simboliza al directorio raíz y que además es el separador de los nombres de directorios y archivos esta representado por un "backslash" (\), en los sistemas basados en Unix , la raíz está representado por el símbolo inverso, un "slash" (/).

Esta sutil diferencia puede confundir e incomodar al usuario en algún momento, mas aún si es principiante, pero es cuestión de tiempo para que uno asimile esta diferencia.

Por ejemplo, si tuviéramos un directorio llamado “personal” y deseáramos acceder desde ambos sistemas, el comando estaría escrito de la siguiente manera :

<i>En DOS</i>	<i>cd \personal</i>
<i>En Linux</i>	<i>cd /personal</i>

En ambos casos obtenemos el mismo resultado pero utilizando una simbología diferente.

Nombres de archivos y directorios

Otro punto donde es necesario reeducarnos es en el tema de los nombres de los archivos y directorios.

Es importante estar plenamente concientes que para Linux una palabra escrita en mayúsculas es diferente a una palabra escrita en minúsculas, aunque posean el mismo contenido, lo que no sucede en DOS. Vale decir que la palabra "orden" es diferente de la palabra "Orden".

Observemos el siguiente ejemplo.

En DOS/Windows estos nombres tienen el mismo significado y son seis maneras de llamar a un solo archivo.

<i>Documento.txt</i>	<i>documento.txt</i>	<i>documento.TXT</i>
<i>Documento.Txt</i>	<i>DocuMentO.txt</i>	<i>Documento.TXT</i>

En Linux estos seis nombres significan seis ficheros diferentes con seis nombres diferentes.

Documento.txt
Documento.Txt

documento.txt
DocuMentO.txt

documento.TXT
Documento.TXT

Cuando menciono la importancia de estar conscientes de estas diferencias, es debido a que para los usuarios de entornos DOS/Windows nos era indiferente almacenar un documento escribiendo su nombre en mayúsculas o en minúsculas, finalmente siempre lo abríamos digitando el nombre, escrito de la manera que sea. En Linux esto no es así de sencillo, el nombre se almacena tal y como el usuario lo digitó y para volver a invocarlo se debe digitar exactamente de la misma manera. Sería lamentable perder tiempo e información por esta leve equivocación.

Es por eso que he tenido el mayor cuidado en escribir los comandos de la manera correcta, adicionalmente a esta consideración, la gran mayoría de comandos e instrucciones en Linux están escritas en minúsculas.

Seguridad

Un concepto al que gran parte de los usuarios de Windows 95 / 98 y DOS no estamos muy familiarizados es el de **seguridad del sistema**. Linux es un sistema operativo "multiusuario" que incorpora un potente sistema de seguridad, no como añadido o extra al paquete original, sino como componente principal de su núcleo y uno de los fundamentos de su filosofía.

La primera demostración de esta seguridad con la que el usuario se topará es con la solicitud de un "login" cada vez que accedemos al sistema.

El "login" esta compuesto por dos objetos, el "nombre de usuario" y la "contraseña". El nombre de usuario es un identificativo único de un usuario en particular ante el sistema, sólo puede existir un usuario con ese nombre

El usuario principal o "súper usuario" recibe de manera automática el nombre de "root". Es éste usuario el encargado de administrar el sistema y también es el único que puede "hacer casi de todo" sin las restricciones habituales de un usuario convencional.

Estructura del directorio raíz (/)

Linux hereda la estructura de directorios del modelo Unix con ciertas modificaciones.

"En Linux, al igual que en Unix, no se accede a los diferentes sistemas de ficheros que el sistema puede usar mediante indentificadores de unidad (como el número de la unidad o su nombre), sino que son combinados en un único árbol jerárquico que representa el sistema de ficheros como una entidad individual."

Linux añade cada sistema de ficheros en su árbol único cuando es montado en un directorio, como por ejemplo /mnt/cdrom. Una de las características más importantes de Linux es su soporte para muchos sistemas de ficheros diferentes. Esto lo hace muy flexible y capaz de coexistir con otros sistemas operativos. El sistema de ficheros más popular de Linux es el EXT2 y éste es el soportado por la mayor parte de las distribuciones de Linux.

Un sistema de ficheros da al usuario una visión ordenada de los ficheros y directorios guardados en los discos duros del sistema, independientemente del tipo de su sistema y de las características de la unidad física subyacente. Linux soporta transparentemente muchos sistemas diferentes (por ejemplo UMSDOS y EXT2) y presenta todos los ficheros y sistemas de ficheros montados como un único árbol virtual. Por esto, los usuarios y los procesos no necesitan generalmente saber en qué tipo de sistema de ficheros está algún fichero, tan sólo lo usan.

Los controladores de unidades por bloques diferencian entre los tipos de unidad física (por ejemplo, IDE y SCSI) y, al menos en lo que concierne al sistema de ficheros, las unidades físicas

son sólo colecciones lineales de bloques de datos. El tamaño de bloque varía entre las unidades; por ejemplo, 512 bytes es una medida común en los "floppy disc", mientras que 1024 son más corrientes en las unidades IDE y, en general, esto es ocultado a los usuarios del sistema. Un sistema de ficheros EXT2 tiene el mismo aspecto, independientemente de la unidad que lo contenga." (David A Rusling)

Además, para poder entender un poco mejor la propia estructura del Linux debemos comprender su distribución.

<i>/bin</i>	Contiene archivos ejecutables públicos
<i>/boot</i>	Directorio donde se localizan los archivos de arranque.
<i>/dev</i>	Directorio donde se ubican todos los dispositivos del sistema
<i>/etc</i>	Configuración de máquina local
<i>/home</i>	Aloja los directorios de los usuarios y servicios como FTP y HTTP.
<i>/lost+found</i>	En este directorio se almacenan los fragmentos de información que han sido recuperados después de una caída del sistema, si los hubiere.
<i>/mnt</i>	Reúne a los dispositivos montados en el sistema, como discos duros, particiones y disquetes.
<i>/proc</i>	Directorio virtual que no existe en el disco de manera original, sino que esta creado en la memoria y su función es la de suministrar información sobre el sistema al núcleo.
<i>/root</i>	Directorio personal del súper usuario.
<i>/sbin</i>	Ejecutables de uso exclusivo del súper usuario, contiene lo necesario para arrancar y montar /usr
<i>/tmp</i>	Directorio temporal de acceso publico
<i>/usr</i>	Directorio de programas compatibles. Esta por lo general puede ser una partición de solo lectura.
<i>/usr/doc</i>	Directorio que contiene toda la información de la distribución (como los Howtos y helps) y de las diferentes aplicaciones instaladas.
<i>/usr/bin</i>	Ejecutables y programas públicos.
<i>/usr/local</i>	Programas instalados para uso local.
<i>/usr/src/linux</i>	Código fuente del Linux.
<i>/usr/doc</i>	Directorio que contiene el grueso de la información de cada programa y comando instalado en el sistema.
<i>/usr/man</i>	Páginas del sistema de ayuda de Linux ("man").
<i>/var</i>	Directorio de archivos de datos variables.
<i>/lib</i>	Contiene exclusivamente bibliotecas necesarias para ejecutar archivos binarios localizados en /bin y /sbin.

Dispositivos del sistema

Se denomina *dispositivo* a todo componente físico del sistema. Estos dispositivos están vinculados al computador por *controladores de dispositivos*, éstos controladores tienen como función brindar una interfase que permita reconocer y controlar adecuadamente un tipo específico de hardware, los cuales están representados por ficheros que están ubicados dentro del directorio */dev*.

Estos *controladores de dispositivos* controlan todo el entorno físico del sistema, desde el teclado, hasta un complejo componente.

Tomemos como ejemplo la primera partición primaria del primer disco duro IDE del computador (lo que normalmente en un entorno DOS sería "C:\"). "C:\\" es la primera partición del primer disco duro, por lo que tenemos dos componentes, la primera partición y el primer disco duro. En Linux esto se representa de manera diferente, al primer disco duro le asignamos el nombre de `/dev/hda` y a la primera partición de `/dev/hda` la llamamos `/dev/hda1`.

En la medida de lo posible los dispositivos reciben nombres en base a las iniciales de su nombre real, por ejemplo `hda` son las iniciales de *Hard Disk A* (primer disco duro).

A continuación una breve lista de los nombres de dispositivos mas utilizados:

<code>/dev/lp</code>	Dispositivo de impresión
<code>/dev/sda</code>	Primer Disco SCSI, primera partición
<code>/dev/fd0</code>	Primer Floppy disk o disquetera
<code>/dev/cdrom</code>	Dispositivo CD-Rom
<code>/dev/ttyS1</code>	Representa al puerto COM2
<code>/dev/ttyS3</code>	Representa al puerto COM4
<code>/dev/cua1</code>	Representa al puerto COM2
<code>/dev/sda4</code>	Cuarta partición de la primera unidad SCSI, también controlador de dispositivo asignado a las unidades ZIP
<code>/dev/hda</code>	Primer disco duro
<code>/dev/hda1</code>	Primera partición del primer disco duro
<code>/dev/hda5</code>	Primera partición extendida del primer disco duro.

Información sobre cada comando

Linux es un sistema operativo especialmente documentado, la mayoría de los comandos tienen sus pantallas de ayuda que brindan una explicación bastante detallada de su utilización.

Para invocar la ayuda desde el prompt, solo debemos digitar:

```
man <comando>
info <comando>
```

Donde `<comando>` es el nombre de la función sobre el cual deseamos mas información.

Tanto el comando `man` como el comando `info` son complementarios, no necesariamente uno posee la información del otro, aunque en algunos casos es así.

Ambos proveen una valiosa fuente de información de primera mano, pero no toda. Dentro del directorio `/usr/doc` existe aún mas información, como los documentos de referencia de cada programa, los *Howto* (*Como*), entre otros.

Para no dejar este tema inconcluso, debemos aclarar que los *howto* (o *como*) son manuales escritos sobre un tema específico y están orientados a brindar al usuario un conocimiento mas profundo sobre ese tema en particular. Existen versiones reducidas o incompletas de los *howto*, denominados *mini-howto* (*mini-como*) que también poseen información valiosa

Finalmente, Linux podrá carecer de ciertas características, pero nunca de falta de información, por el contrario, esta mucho mas documentado que otros sistemas operativos, es mas, existe toda una organización abocada a la producción de información sobre Linux. Este grupo de personas e instituciones conforman el **LDP** (*Linux Document Project*) y uno de sus representantes mas populares en nuestro medio es *LuCAS* (*Linux en CASTellano*).

Distribuciones de Linux

Linux es un sistema amparado bajo licencia GNU que permite su libre distribución, así también como su distribución comercial. Al ser distribuido con el código fuente, éste es manipulado, perfeccionado, complementado, documentado y finalmente distribuido, es de allí que nacen las *distribuciones*.

Cada distribución presenta un código modificado, pero compatible con aplicaciones Unix y otros núcleos Linux, de manera que el programa que es leído en Red Hat, pueda ser igualmente reconocido en Debian.

Existen varias distribuciones disponibles, pero las mas reconocidas son :

Red Hat	Caldera
Debian	Slackweare
Suse	Umsdos

Fichero o archivo

Linux incluye el concepto de "fichero" como parte de su filosofía. Un fichero es aquel que representa a "algún componente de nuestro sistema", sea físico (hardware) o lógico (software)".

Por ejemplo, un "dispositivo" como el modem esta representado por el fichero `/dev/ttyS2`. Cada vez que debamos utilizar el modem, haremos referencia a ese fichero.

Es realmente sencillo confundir fichero con archivo. Archivo es aquel que existe físicamente en el disco duro, ya que ocupa una porción de disco, en cambio, un dispositivo puede estar representado por un fichero que vemos como archivo pero que realmente no existe como tal en nuestro disco.

Un ejemplo de lo explicado anteriormente es el directorio `/proc` el cual se tratará en las líneas siguientes.

Ejecutar un script

Un script no es otra cosa que un conjunto de comandos ordenados lógicamente para cumplir un fin específico, ya que nos permite ejecutar múltiples y diferentes órdenes con un solo comando.

Es mas que seguro que en algún momento nos veremos en la necesidad de realizar nuestros propios scripts para agilizar y automatizar algunas de las tareas que nos requiere el sistema. Muchos de los que crecimos con sistemas operativos como el DOS y el Windows en su momento vimos y trabajamos con los archivos BAT. Este archivo no es mas que un script para DOS.

Los scripts en Linux no necesitan una extensión específica para poder ejecutarse, solo requieren del premiso necesario que lo convierta en *ejecutable* y de ser el caso, incluirle la cabecera de ejecutable.

Cuando deseemos realizar un script, sugerimos colocar como primera línea la siguiente cadena :

```
#!/bin/sh
```

En algunos casos necesitaremos anteponer los símbolos "." (sin las doble comillas) para poder *ejecutar* un script desde el mismo directorio.

Sobre las versiones de Linux

Cada versión del núcleo viene debidamente numerada, esta numeración obedece a un orden simple. Una versión del núcleo considerada *estable* posee un identificador "par" en el medio de la cadena (2."0".30), una versión *en desarrollo* (o inestable) tiene el identificador del medio como "impar". (2."1".30).

Para instalar un archivo RPM

Los archivos "RPM" son paquetes que contienen un programa o componente para instalar en nuestro sistema. Esta instalación se puede hacer desde Xwindows o desde el prompt. En este último caso se hace de la siguiente manera:

```
rpm -Uvh <paquete.rpm>
```

Consolas Virtuales

Una muestra de la capacidad "multitarea" que caracteriza a Linux son las *consolas virtuales*, que no son mas que diferentes pantallas realizando procesos diferentes utilizando un mismo núcleo.

Por defecto nos aparece la "primera consola virtual" cuando iniciamos Linux, pero podemos cambiarnos de consola con solo apretar una de las siguientes combinaciones de teclas:

[ALT]+[F1]	Primera consola virtual.
[ALT]+[F2]	Segunda consola virtual.
[ALT]+[F3]	Tercera consola virtual.
[ALT]+[F4]	Cuarta consola virtual.
[ALT]+[F5]	Quinta consola virtual.
[ALT]+[F6]	Sexta consola virtual.

Podemos tener las seis consolas abiertas de manera simultanea, sin que una interfiera con la otra. También sería una solución practica un cambio de consola cuando el sistema se nos estanca por algún motivo sin vernos obligados al *reset* o en caso de estar haciendo un *backup* en primer plano, simultáneamente trabajar en otra consola.

Haciendo mas fácil el trabajo – Creando Enlaces simbólicos

Por la propia estructura de archivos del Unix, se hace necesario *acercar* los ficheros de trabajo lo mas posible al usuario, esta es la razón de ser de los "enlaces simbólicos"

Un enlace simbólico (o enlace blando) no es otra cosa que un pequeño archivo que "apunta" hacia un directorio o archivo ubicado generalmente en otro lugar, el cual puede utilizarse para acceder directamente a este archivo o directorio.

Es conveniente utilizar "enlaces simbólicos" cuando el documento o directorio al cual queremos hacer referencia esta un poco "lejos" de nuestra ubicación actual.

De acuerdo a la distribución de Linux o al modo en que nos encontremos trabajando podemos visualizar de una u otra manera a un enlace simbólico. Por ejemplo, en Linux Red Hat, al efectuar

un "ls -l", veremos una lista de los archivos y directorios en donde los enlaces tendrán un símbolo "->" en donde aparecerá el archivo o directorio al que apunta.

```
-rwxrwx--- 1 fernando fernando 63214 oct 26 18:48 Comandos1
-rw-r-xr-x 1 fernando fernando 66866 oct 27 11:39 Comandos2
-rw-r-xr-x 1 root root 66744 oct 27 11:34 Comandos3
lrwxrwxrwx 1 root root 13 oct 27 16:38 almacen -> /tmp
```

El ejemplo anterior tenemos tres archivos y un enlace simbólico llamado "almacén" que realmente apunta al directorio */tmp*. Este fichero sólo contiene dentro de él la manera de cómo llegar al */tmp*

En algunos programas, como el "Midnigh Commander" (mc) un enlace a un directorio se visualizan con el símbolo ~ al final del nombre del enlace, en cambio el enlace a un archivo será con la @ antes del nombre, pero a ambos se les llama solo por el nombre.

ej.	raul~	Enlace a Directorio
	@raul	Enlace a archivo

Para crear un enlace simbólico "blando":

```
ln -s <fichero / directorio> <nombre_del_enlace>
```

Ejemplo:

```
ln -s /usr/doc/samba/HOWTO/HOWTO howto
```

Ahora puede referirse al mismo documento de dos maneras :

```
"howto" ó "/usr/doc/samba/HOWTO/HOWTO"
```

Un enlace simbólico es una puerta de entrada directa a un fichero en particular, si esta enlazado a un directorio podemos manejarlo con "cd"

Para eliminar un "enlace" se utiliza "rm", hay que tener en cuenta que borrar el enlace no afecta el archivo original a menos que se utilice el parámetro "-f".

Nota : Todo texto que va entre los símbolos <> debe ser sustituido por su valor real (sin los <>). Cada comando posee una descripción genérica de su función, para mayor referencia consulte el comando "man" o los "howto" sobre el tema.

Creación de alias para nuestro sistema

Un *alias* no es otra cosa que un sobrenombre para una computadora, toda esta información se almacena en el archivo */etc/hosts*.

Ej.

#IP MAQUINA	NOMBRE REAL	ALIAS
#-----	-----	-----
127.0.0.1	localhost	localhost.localdomain
192.168.100.1	servidor.principal	servidor ffg.net
192.168.100.2	pc001.principal	contabilidad
192.168.100.3	pc002.principal	administracion

Creación y control de usuarios

Después de instalado el Linux, el primer paso que debemos dar es la creación de por lo menos un usuario (además de root), el cual usaremos para nuestro trabajo. Es conveniente utilizar al súper usuario (root) para labores de configuración y administración del sistema y no para labores cotidianas.

Existen dos maneras de crear un usuario, básicamente son lo mismo pero difieren en el modo de asignar las contraseñas.

1) Contraseña personalizable

<code># adduser <usuario></code>	añade al usuario especificado en <usuario>.
<code># passwd <usuario></code>	asigna contraseña
<code># chfn <usuario></code>	Ingresa información básica del usuario
<code># usermod -G <grupo> <usuario></code>	Cambia de grupo a un usuario

2) Contraseña No personalizable

<code># adduser <usuario></code>	añade al usuario especificado en <usuario>.
<code># mkpasswd <usuario></code>	genera una contraseña alfanumérica y la introduce en el registro del usuario.
<code># chfn <usuario></code>	Ingresa información básica del usuario
<code># usermod -G <grupo> <usuario></code>	Cambia de grupo a un usuario

Con el método anterior, el sistema nos muestra por única vez la contraseña que asigno a un usuario, la anotamos y la entregamos.

Nota : Desde el XWindows se puede utilizar el comando "kuser" para lograr una mas eficiente administración de los usuarios y sus quotas.

A continuación describimos de manera genérica todos los comandos con que permiten añadir, modificar o eliminar información de los usuarios y sus grupos. (incluidos los anteriormente mencionados)

Finalmente, tal vez Ud. prefiera utilizar el comando "userconf".

<code>userconf</code>	Potente herramienta que permite ingresar toda la información concerniente a las cuentas de usuarios.
-----------------------	--

Comandos de control

De usuario

<code>id <usuario></code>	Muestra información del usuario, así como UID, GUD y todos los grupos a los que pertenece.
<code>adduser <usuario></code>	añade a <usuario> a la base de datos del sistema
<code>userdel</code>	Elimina a un usuario
<code>usermod -G <grupo> <usuario></code>	Cambia de grupo a un usuario
<code>chfn <usuario></code>	Ingresa información básica del usuario
<code>usermod</code>	Cambia la información de un usuario. Este comando es bastante poderoso, ya que permite cambiar el UID del usuario.

userinfo ejemplo :
usermod -u 900 fernando
(con esto se me asigna un nuevo UID)
Muestra información del usuario en X

De grupo

groupadd Crea un nuevo grupo de usuarios
groupdel Elimina un grupo de usuarios
groupmod Modifica información de un grupo de usuarios
grpck Verifica la integridad de los archivos de grupos
groups Muestra los grupos existentes

Otros relacionados

chgrp Cambia a un usuario de grupo.
(Ej. *chgrp* <grupo> <usuario>)
groupmod Modifica información de un grupo de usuarios
chsh permite modificar el shell por defecto
chown Asigna y cambia propiedad de un directorio / archivo a un usuario y/o grupo.
gpasswd Administra el archivo */etc/group*
finger Se puede visualizar todos los datos introducidos con el comando "chfn". (Ej. *finger fernando*)
who visualiza a todos los usuarios conectados al terminal

Inhabilitar acceso a un usuario

Si trabajamos Linux en un entorno de red y como servidor, nos será necesario el restringir al máximo el acceso de usuarios al equipo, para eso existe un método bastante sencillo para impedir que usuarios accedan al servidor y este es cambiándoles el "shell" por defecto ("/bin/sh"). Para ello ejecutamos :

```
chsh <usuario>  
(y luego ingresamos) /bin/false
```

Con este procedimiento el usuario <usuario> tendrá todos sus privilegios vía red, pero no podrá realizar un login directamente al computador y trabajar de manera interactiva o directa con el.

Para devolverle la autoridad para autenticarse y ser aceptado por el servidor de manera interactiva en el servidor ejecutar :

```
chsh <usuario>  
(y luego ingresamos) /bin/bash
```

Si deseamos restringir al máximo el acceso de los usuarios al servidor, debemos permitir solo conexiones FTP anónimas, incluyendo el nombre de cada usuario en el archivo */etc/ftpaccess*.

Asignación de cuotas en el servidor

Las cuotas son el instrumento que permiten controlar de manera eficiente la cantidad de espacio que un usuario puede utilizar en una "partición" específica.

Las "cuotas" se asignan por particiones, no por directorios. Así, si tenemos una partición dedicada exclusivamente para los directorios /home, podemos controlar el máximo de espacio que un usuario pueda utilizar y este puede ser diferente al utilizado en otra partición con otra finalidad (Ej., /tmp).

La manera de asignarlo es mediante el "linuxconf", opción "quotes", previamente en la sección "particiones", hay que activar por cada partición que deseemos controlar las cuotas.

Estas están expresadas en kilobytes (Kb).

Comandos :

quota	Permite asignar una cuota de espacio en el servidor a cada usuario sin que este la pueda sobrepasar.
quotaon	Activa el servicio de cuotas
quotaoff	Desactiva el servicio de cuotas.
edquota	Modifica las cuotas, se necesita saber manejar el editor <i>vi</i>
repquota <dev>	Visualiza el estados de las cuotas, sea de usuarios o de grupos. Ej. repquota /dev/hda8 -u

Cabe señalar que las cuotas pueden asignadas ser con un valor individual y diferente por cada usuario. Del mismo modo a los grupos.

Asignación de seguridad en los directorios y archivos

Uno de los pilares de la seguridad de Linux es el nivel de protección que asigna cada objeto, como archivo o directorio en particular.

Cuando ingresamos como *root* al sistema (o como un usuario cualquiera) y creamos nuestra estructura de trabajo personalizada para compartir con otros usuarios nuestros archivos, no nos damos cuenta que como *creadores* de esos directorios el sistema nos hace automáticamente propietarios de los mismos y que el resto de los usuarios no podrán acceder totalmente a menos que nosotros lo permitamos.

Para eso existen una serie de utilidades en X-Windows, pero ninguna mejor y mas practica que el modesto comando *chmod*.

Este comando permite asignar la seguridad a todos los archivos y directorios del sistema.

Para poder continuar, es importante conocer que Unix reconoce tres tipos "diferentes" de individuos a los que puede asignar diferentes permisos:

- 1) El propietario (u)
- 2) El grupo propietario del directorio (g)
- 3) El resto de usuarios excluidos de los dos primeros. (o)

Para ver los propietarios de cada archivo ejecutar "ls -l" o "vdir".

Cada permiso tiene un valor único, aquí sus equivalencias:

	Derecho	Valor
Set UID	(ver cuadro inferior)	+4000
Set GUD	(ver cuadro inferior)	+2000
Sticky bit	(ver cuadro inferior)	+1000
Usuario propietario	Solo lectura (r)	+400

Usuario propietario	Escritura (w)	+200
Usuario propietario	Ejecución / búsqueda (x)	+100
Grupo propietario	Solo lectura (r)	+40
Grupo propietario	Escritura(w)	+20
Grupo propietario	Ejecución / búsqueda(x)	+10
Otros	Solo lectura (r)	+4
Otros	Escritura(w)	+2
Otros	Ejecución / búsqueda (x)	+1

Set UID	Fichero que al ejecutarse lo hace con privilegio del usuario a quien pertenece el UID (generalmente root).
Set GUI	Fichero que al ejecutarse lo hace con el privilegio propio al grupo que pertenece el GUI.
Sticky	

Son 12 permisos diferentes que se pueden combinar para conseguir una restricción.

Podríamos resumir valores comunes y estándar de la siguiente manera :

Tipo de acceso	Protección	Valor
Solo Usuario propietario	Todos los permisos	700
Usuario y Grupo propietario	Todos los permisos	770
Documentos protegidos contra el grupo	Solo lectura para el grupo	744
Script Ejecutable	Lectura / Ejecución	755
Todos los permisos	Lectura Escritura / Ejecución	777
Archivos protegidos contra escritura	Solo lectura y ejecución para todos	555

Para visualizar en pantalla todos los archivos y sus respectivos permisos, propietarios, digitar : "ls -l -v" o "vdir". Si se desea dar salida a una impresora añadir |lpr (ej. "ls -l |lpr" o "vdir |lpr") o "vdir >listado.txt hacia un archivo.

También se puede asignar o quitar permisos de la siguiente manera :

```
chmod <grupo> <+|-> <permiso(s)> <archivo(s)>
```

Los permisos son (r) Lectura, (w) escritura y (x) ejecución.

<i>chmod u -r <archivo></i>	Quita el derecho de lectura al usuario
<i>chmod g -r <archivo></i>	Quita el derecho de lectura al grupo
<i>chmod o -r <archivo></i>	Quita el derecho de lectura al resto (owner)
<i>chmod ugo+rwx <archivo></i>	Asigna todos los permisos.

Las opciones de este comando son:

- R Incluye todos los subdirectorios y ficheros. (recursivo)
- f Fuerza el cambio, aún si los ficheros no le pertenecen.
- v Muestra los mensajes de error.

En caso de que deseemos asignarle la administración de un directorio que no sea del sistema a alguno de estos usuarios, debemos utilizar el comando chown :

chown usuario.grupo /directorio -R

ej.

chown fernando.fernando /intranet -R

Con el ejemplo anterior, he asignado al usuario fernando del grupo fernando la propiedad y control del directorio /intranet y todos sus directorios. Se debe ser el propietario del directorio (o root) para asignar estos privilegios.

<i>umask</i>	Asigna todos los permisos. Modifica la mascara por defecto en el momento de crear un archivo. Es recomendable establecer este valor a 000 (ej. umask 000)
--------------	--

Montaje de unidades de disco

Una de las características del Linux es que todas las unidades se deben de montar al sistema para que sean reconocidas.

Los comandos que permiten el montaje de unidades de disco, son :

<i>mount</i>	Monta una unidad de disco en una división lógica
<i>umount</i>	Desmonta una unidad montada (ej. umount /mnt/cdrom)

En los siguientes ejemplos veremos como montar y desmontar una unidad de disco. Generalmente las unidades montadas se encuentran en /mnt pero pueden estar en cualquier sitio, siempre y cuando el directorio de montaje exista.

montar una particion FAT en /mnt/disco_fat	mount -t vfat /dev/hda5 /mnt/disco_fat
montar un disquete DOS como "solo lectura"	mount -t umsdos /dev/fd0 /mnt/disquette
montar un CD-ROM	mount -t iso9660 /dev/cdrom /mnt/cdrom

Nota : Para desmontar una unidad, esta no debe estar siendo utilizada o bloqueada por ningún otro servicio, sin importar si se está utilizando o no.

Montaje de unidades de red con samba (Compartiendo recursos)

Este procedimiento es para que Linux pueda acceder a unidades compartidas en otras máquinas de la red con sistema operativo Windows, no a la inversa. Para ver cómo Linux comparte sus recursos, revisar la sección dedicada a *Samba*.

Al igual que Windows puede asignar una unidad lógica a un directorio compartido vía red, Linux puede asignar un directorio al mismo recurso compartido.

Por ejemplo, tenemos un recurso compartido en "\\auxiliar\shared", para montarlo utilizaríamos el comando "smbmount" de la siguiente manera:

```
smbmount //auxiliar/shared /mnt/auxiliar_shared -N
```

Donde `//auxiliar/shared` es el recurso que se encuentra compartido en la red y `/mnt/auxiliar_shared` es el directorio donde vamos a montarlo dentro de nuestro sistema. El parámetro `-N` evita que el sistema nos pida contraseña. (Esto último siempre y cuando el recurso no esté protegido con contraseña).

Para *desmontar* un directorio montado con *smbmount*, utilizar el comando *smbumount*

```
sbumount /mnt/auxiliar_shared
```

Para conocer los recursos compartidos de una máquina en particular, utilizar

```
smbclient -L \\<máquina>
```

Es importante saber que en Windows 9x sólo se permite compartir en tres modos:

"Total", "Solo lectura" y "Dependiendo de la contraseña"

Con los dos primeros, *smbmount* no tiene ningún problema (incluso se puede incluir el parámetro `-N` al realizar la conexión), pero en el último modo, *smbmount* solo puede conectarse con la contraseña de *solo lectura* (introducida manualmente, obviando `-N`), mas no con la contraseña de "acceso total".

CAPITULO II

COMANDOS LINUX

CAPITULO II

COMANDOS LINUX

Las siguientes páginas contendrán una lista de los comandos mas comunes con los que un usuario se pueda topar, además se incluyen varios que pueden resultar de utilidad en un momento determinado. La mayoría de los mismos no contendrán sintaxis sino una breve descripción de su función.

Es importante recordar que estos comandos pueden variar según la distribución con que se trabaje, en este caso todos han sido tomados de la versión 6.1 de Red Hat, base de este escrito.

Utilidades de configuración

Existen diversos comandos que nos permiten configurar apropiadamente el Linux.

a) Principales

<i>linuxconf</i>	Permite configurar casi todo el sistema de manera bastante sencilla.
<i>userconf</i>	Permite configurar las cuentas de usuarios de manera gráfica
<i>netconf</i>	Permite configurar la mayoría de parámetros de la red
<i>mailconf</i>	Configura el sendmail.
<i>fsconf</i>	Configurador de 'file systems'
<i>xconf</i>	Configura el sistema X.
<i>authconfig</i>	Permite activar y configurar servicios como NIS, Shadow passwords y MD5 passwords.
<i>kbconfig</i>	Configura el idioma del teclado
<i>sndconfig</i>	Configura el dispositivo de sonido
<i>timeconfig</i>	Configura la zona horaria
<i>mouseconfig</i>	Configura el mouse
<i>Xconfigurator</i>	Permite configurar el entorno X.

b) Auxiliares

<i>control-panel</i>	Ejecuta el panel de control en X.
<i>pws</i>	Configura HTTP bajo X.
<i>kppp</i>	Configura acceso telefónico a redes.
<i>kuser</i>	Utilidad grafica de X que permite realizar de manera cómoda toda la gestión de usuarios.

Utilidades de diagnostico

<i>e2fsck</i>	Analiza partición ext2 de Linux.
---------------	----------------------------------

<i>free</i>	Muestra la cantidad de memoria libre
<i>hlttest</i>	Utilidad que permite testear el teclado.
<i>hwdiag</i>	Utilidad de diagnostico del hardware conectado a nuestra computadora.
<i>hostid</i>	Visualiza el identificador numérico del host
<i>ipcs</i>	Provee información sobre el sistema IPC
<i>ktop</i>	Otro medidor grafico de recursos del sistema
<i>kpm</i>	Medidor de recursos grafico
<i>lsthome</i>	Muestra todos los directorios "home" actualmente asignados a usuarios.
<i>listalias</i>	Muestra los "alias" registrados en el sistema
<i>logname</i>	Muestra el nombre de usuario con login actual.
<i>ps -aux</i>	Pone en pantalla todos los procesos que ocupen memoria. (ej. <i>ps -aux >memoria.txt</i>)
<i>showkey</i>	Muestra el valor hexadecimal de una tecla
<i>tack</i>	Utilidad de test de terminal
<i>top</i>	Visor de recursos del sistema en modo texto.
<i>vmstat</i>	Revisa estado de la memoria virtual

Comandos de gestión de archivos y directorios

<i>~/</i>	Alias para nuestro directorio personal. No es un comando
<i>></i>	Redirecciona una salida. (ej. " <i>ls -F >listado.txt</i> " Esto crea un archivo llamado "listado.txt" con el contenido del directorio.). No es un comando
<i>cd</i>	Cambia de directorio (ej, <i>cd /</i>)
<i>cksum</i>	Genera / Analiza un identificador de cada archivo basado en la cantidad de bytes del archivo y de un CRC. Programa bastante útil para verificar la autenticidad de un archivo.
<i>cmp <fich1><fich2></i>	Compara 2 ficheros.
<i>comm <fich1><fich2></i>	Compara dos ficheros ordenados, fila por fila
<i>cp</i>	Copia archivos (ej. <i>cp /tmp /home</i>)
<i>date</i>	Muestra la hora y fecha de sistema.
<i>dir</i>	Muestra nombres de ficheros y archivos
<i>diff <fich1> <fich2></i>	Busca diferencias entre dos ficheros.
<i>diff3 <f1><f2><f3></i>	Busca diferencias entre tres ficheros.
<i>df</i>	Muestra el espacio disponible / usado en cada partición. Este comando es bastante útil para administrar el espacio de disco.
<i>du</i>	Muestra el espacio de disco ocupado por cada archivo. (ej. " <i>du -s <dir></i> " muestra el total ocupado por un directorio. " <i>du -b</i> " muestra el detalle en bytes)
<i>efax</i>	Envía y recibe faxes.
<i>fsck</i>	Chequea y repara el sistema de archivos Linux.
<i>git</i>	Otro file manager en modo texto.
<i>ln -s <fichero></i>	Crea un "vinculo simbólico" a un archivo. (ej. " <i>ln <fichero a vincular> <nombre _ vínculo></i> ")
<i>ls</i>	Muestra nombres de ficheros y archivos. Se utiliza con parámetros según los resultados que deseemos. Por ejemplo " <i>ls -l</i> " muestra los nombres de los archivos, sus propietarios y

<i>mc</i>	<p>Midnigth Commander, un "file manager" bastante completo y practico, entre sus características esta el permitirnos leer ficheros RPM e instalarlos.</p>
<i>mkdir</i>	Crea un directorio.
<i>pwd</i>	Muestra la ruta completa del directorio actual.
<i>pstree</i>	Muestra un árbol de procesos.
<i>pilot</i>	Gestor de ficheros similar al Pine.
<i>reset</i>	Realiza un reset del sistema
<i>rm</i>	Remueve archivos y directorios (ej. <code>rm /temp/* -R</code>)
<i>stat <archivo></i>	Muestra en pantalla la información básica del archivo al que se hace referencia, como el tipo de archivo, a quien pertenece, a que grupo pertenece, su tamaño, el tipo de archivo que es, los permisos que tiene asignados, los vínculos que tiene sobre el, las tres fechas básicas (acceso, modificación y cambio)
<i>su</i>	Sirve para iniciar una sesión con privilegios de <i>súper usuario</i> dentro de otra sesión. Requiere la contraseña de <i>root</i> .
<i>tree</i>	Visualiza el árbol de directorios y archivos del directorio actual.
<i>touch</i>	Cambia la fecha y hora del archivo.
<i>vdir</i>	Visualiza en pantalla los permisos y propietarios de los archivos seleccionados.

Gestión de audio y multimedia

<i>aumix</i>	Personaliza el volumen de los diversos dispositivos de audio, como el CD-Rom, Wave, Midi. etc. Diseñado para trabajar desde la consola.																				
<i>cdp</i>	<p>CD-Player interactivo para consola. requiere que el <i>NumPad</i> este apagado para controlar el programa:</p> <table> <tr><td>9</td><td>Play</td></tr> <tr><td>8</td><td>Pausa/resumen</td></tr> <tr><td>7</td><td>Stop</td></tr> <tr><td>6</td><td>Siguiente</td></tr> <tr><td>5</td><td>Replay</td></tr> <tr><td>4</td><td>Anterior</td></tr> <tr><td>3</td><td>Adelantar 15 segundos</td></tr> <tr><td>2</td><td>Abortar</td></tr> <tr><td>1</td><td>Retroceder 15 segundos</td></tr> <tr><td>0</td><td>Salir pero la música continua</td></tr> </table>	9	Play	8	Pausa/resumen	7	Stop	6	Siguiente	5	Replay	4	Anterior	3	Adelantar 15 segundos	2	Abortar	1	Retroceder 15 segundos	0	Salir pero la música continua
9	Play																				
8	Pausa/resumen																				
7	Stop																				
6	Siguiente																				
5	Replay																				
4	Anterior																				
3	Adelantar 15 segundos																				
2	Abortar																				
1	Retroceder 15 segundos																				
0	Salir pero la música continua																				
<i>cdplay</i>	CD-Player no-interactivo, toca el cd de principio a fin Se detiene con "eject"																				

Comandos - utilidades misceláneas

<i><proceso> &</i>	Coloca un proceso o programa en <i>background</i>
<i>at</i>	Coloca ordenes en cola para ejecución en un

	momento determinado. ejemplo at now + 1 day <<EOF >shutdown -r now >[CTRL]+[D]
<i>atd</i>	Corre trabajos postergados para ejecución posterior
<i>atq</i>	Lista todos los trabajos pendientes de at
<i>atrm <#cola></i>	Borra un # de cola de at. El valor del #cola se obtiene con "atq"
<i>audiocompose</i>	Permite grabar audio desde pantalla en modo texto.
<i>audiosend</i>	Envía mail con audio
<i>bg</i>	Cambia entre foreground a background.
<i>cat <archivo></i>	Muestra el contenido de un archivo. Este comando repite las cadenas escritas en el prompt. Se puede usar como un editor rudimentario digitando "cat > archivo.txt", escribir y [CTRL]+[D] para finalizar.
<i>cal <mes><año></i>	Muestra un "CALendario" con los datos indicados, si no se le suministra datos, muestra el mes y el año actuales.
<i>clear</i>	Borrar la pantalla.
<i>comp</i>	Compone un mensaje de correo electrónico.
<i>compress</i>	Comprime y descomprime data. Le añade extensiones "Z" a todos los archivos comprimidos, no los añade en un único archivo. Para descomprimir se utiliza <i>uncompress * -r</i> del mismo modo para comprimir recursivamente de usa <i>compress * -r</i>
<i>crontab</i>	Monta un fichero individual del cron por cada user.
<i>dialog</i>	Utilidad que permite crear cajas de dialogo, mensajes opciones de si y no, etc. Sus opciones son : --yesno "texto" "alto" "ancho" --msgbox "texto" "alto" "ancho" --infobox "texto" "alto" "ancho" --inputbox "texto" "alto" "ancho" <init> --textbox "texto" "alto" "ancho" entre otras... (man dialog) ej. dialog "Hola mundo..." "5" "40"
<i>dos</i>	Carga un emulador DOS en memoria
<i>eject</i>	Comando que tiene como función expulsar o retraer (si es posible), la bandeja de una unidad extraíble. De modo natural expulsa la bandeja del CD, con el parámetro "-t" retrae la misma bandeja. También puede ser usado para unidades extraíbles como una Zip ("eject /dev/zip"). Nota: El retraer la bandeja del CD no lo monta.
<i>expr</i>	Permite realizar un calculo (ej. "expr 1 + 1" = 2)
<i>expect</i>	Permite revisar el historial de comandos ejecutados desde un prompt. Dir es un buen punto para comenzar.
<i>expectk</i>	Expect para X.
<i>fgconsole</i>	Imprime en pantalla en numero de consola activa.
<i>ftdump</i>	Información sobre fuentes True Type.
<i>fdformat</i>	Utilidad de formateo de discos.
<i>fg</i>	Coloca en primer plano un proceso de

<i>find <dir><expresion></i>	<i>background</i> y viceversa. Busca una expresión en los ficheros del directorio indicado. Ejemplos: find /etc -name *ftp* find / -name *fernando*
<i>grep <cad.><fichero></i>	Permite localizar una cadena en particular dentro de un fichero y extrae las coincidencias en pantalla.
<i>gmc</i>	Explorador en X.
<i>gpm</i>	Carga el dispositivo del mouse en modo texto.
<i>gnorpm</i>	Manejador de instalaciones RPM de Gnome.
<i>gnuchess</i>	Ajedrez en modo texto.
<i>head <fichero></i>	Muestra las primeras 10 líneas de la cabecera del <i>fichero</i> .
<i>hpcdtoppm</i>	Convierte un archivo Photo CD en una imagen.
<i>jobs</i>	Lista los procesos en <i>background</i>
<i>kfloppy</i>	Formateador de disquetes.
<i>kfmsu2</i>	Explorador o File manager del súper usuario
<i>kill %<proceso></i>	Elimina un proceso en "background". El valor del proceso se obtiene con "jobs". (ej. "kill %1")
<i>less <fichero></i>	Permite visualizar cómodamente un archivo en pantalla. Similar al comando <i>type</i> del DOS
<i>logname</i>	Muestra nuestro logname (nombre de login)
<i>mail <usuario></i>	Permite enviar un correo a un usuario interno. ej. mail root Subject : Test Prueba de correo [CTRL]+[D]
<i>man <comando></i>	Utilidad de información de cada comando, nos muestra el contenido de su página de ayuda.
<i>more <fichero></i>	Muestra el fichero de manera que podamos leerlo por pantallas. Si se desea ver mas líneas, incluir el parámetro -l (ej. "head -l 15 <fichero>")
<i>mkbootdisk</i>	Crea un disco booteable
<i>pine</i>	Gestor de correo nativo con algunas prestaciones.
<i>rundos</i>	Emulador DOS
<i>shutdown</i>	Apaga la maquina, descargando primero los demonios o servicios. Su opción mas común es: "shutdown -r now". Pero si se desea temporizar el apagado, se puede usar: # shutdown 20:00 & # why? EL SERVIDOR SE APAGARA A LAS 20:00 ó # shutdown -r now # apaga el sistema YA!!
<i>tail <fichero></i>	Función gemela a la de "head", solo que este muestra las últimas líneas.
<i>uptime</i>	Informa sobre el tiempo en que el sistema ha estado activo, cuanto tiempo ha transcurrido desde que se encendió por ultima vez y la cantidad de usuarios enganchados.
<i>who</i>	Muestra todos los usuarios activos en el sistema

<i>w</i>	Informa los usuarios actuales en el sistema y que están haciendo.(ej. "w root" - "w -f" para sistemas remotos)
<i>wc <archivo></i>	Obtiene la cantidad de líneas, palabras y bytes de un archivo. Sus parámetros son "-c" para la cantidad de bytes, "-w" para la cantidad de palabras y "-l" para la cantidad de líneas. Utilizado sin parámetros devuelve los tres valores.
<i>write <usuario></i>	Escribe en la consola de >usuario> todo lo que se escriba en nuestra consola, para que se convierta en una especie de "chat" el otro usuario debe escribir <i>write <tu_usuario></i> . No funciona con "root". Para finalizar, generar u EOF con [CTRL]+[d] en ambas consolas.
<i>writetv <tty><mensaje></i>	Inserta <mensaje> en la línea de comandos de /dev/ttyx ej. <i>write /dev/tty2 ls -l ig*</i>
<i>xman</i>	Manuales en X
<i>xmms</i>	XMultimedia system.
<i>xplorer</i>	Explorador con semejanza al de Windows

Imprimir desde el prompt

Para los que alguna vez utilizamos DOS es fácil recordar el comando PRINT x, en las siguientes líneas averiguara cual es el comando Linux para imprimir.

<i>lpr <archivo></i>	Donde <archivo> es el documento que desea imprimir. ejemplo. <i>lpr fichero.txt</i> (envía a la cola de impresión el archivo "fichero.txt")
<i>lpq</i>	Examina los trabajos que están en la cola de impresión.
<i>lprm</i>	Remueve los trabajos en cola de impresión.
<i>lpc</i>	Programa de control de impresión. Solo accesible como súper usuario.
<i>lptest</i>	Envía una cadena de caracteres de prueba a la impresora por defecto.
<i>lpunlock</i>	Desbloquea una impresora bloqueada.
<i>pac</i>	Informa sobre el dispositivo de impresión utilizado en ese momento.

Comandos de red

<i>dig</i>	Envía una búsqueda de dominio al servidor de nombres.
<i>ftp</i>	Realiza una conexión FTP con otra maquina. =====
	ftp> open 192.168.100.1
	ftp> user: xxxxx
	ftp> password : xxxx
	ftp> dir
	ftp> get <fichero> ; copia fichero remoto a PC
	ftp> put <fichero> ; copia fichero a PC remota.
	ftp> close
	=====

<i>ftpcount</i>	Cuenta los usuarios conectados por FTP.
<i>ftpwho</i>	Muestra quienes son los usuarios conectados al FTP.
<i>finger</i>	Visualiza información de un usuario.
<i>gftp</i>	FTP bajo entorno X
<i>lynx <url></i>	Navegador WWW en modo texto.
<i>nmbd</i>	Demonio que provee el nombre NetBIOS y da soporte de navegación. Depende del archivo <i>/etc/smb.conf</i> .
<i>pwck</i>	Verifica la integridad de los archivos de passwords.
<i>rlogin <host></i>	Login remoto.
<i>rusers</i>	Ver quién esta conectado a la red.
<i>rmail</i>	Acceso remoto a correo electrónico.
<i>rz</i>	Permite recibir vía dial-up archivos.
<i>rx</i>	Permite enviar o establecer conexión pc-pc via modem
<i>rwho</i>	Permite ver quien esta conectado a la red.
<i>rsh</i>	Activa un shell remoto
<i>smbclient -M host</i>	Envía un mensaje por red a un host.
<i>smbstatus</i>	Muestra el estado del samba y los recursos bloqueados
<i>smbrun</i>	Aplicación que facilita la ejecución remota de programas.
<i>smbd</i>	Activa los servicios de compartir por red, los archivos y dispositivos de impresión
<i>telnet <ip></i>	Acceso remoto a través de la red
<i>wall <mensaje></i>	Envía un mensaje a todas las consolas activas

Algunos comandos "samba" están mas detallados en la sección que trata sobre samba.

Compresión y descompresión de archivos

Si bien el comando "tar" es el de mas difusión y uso en Linux, debemos conocer que la eficiencia de este programa usado individualmente, es ineficiente para fines de *compresión* de información, ya que este no comprime, solo empaqueta. por eso se hace necesario utilizarlo con otros programas que si cumplen con la función de "comprimir".

<i>tar</i>	Para comprimir : tar -Mcvf <nombre_archivo_a_generar> <dir/> Para descomprimir : tar -Mxpvf <archivo_a_restaurar>
------------	--

Hay que tener en cuenta en el momento de descomprimir que si se utilizan los parámetros arriba mencionados la descompresión se efectuará en el directorio donde nos encontremos, a menos que indiquemos el "directorio de destino". (El parámetro "M" solo sirve si es "multivolumen")

<i>compress</i>	Comprime y descomprime data. Le añade extensiones "Z" a todos los archivos comprimidos, no los añade en un único archivo. Para descomprimir se utiliza <i>uncompress * -r</i> del mismo modo para comprimir recursivamente de usa <i>compress * -r</i>
<i>gzip</i>	-d descomprime -v comprime -l lista -R Recursivo

El Midnight Commander (mc)

El Midnight Commander es un programa que integra muchas capacidades para el manejo y administración de ficheros y directorios en sistemas operativos basados en Unix.

Se orienta básicamente a usuarios de consolas (modo texto), aunque también se puede utilizar desde el XWindows (con ciertas limitaciones).

Básicamente comprende de :

- * Un sistema interactivo bastante completo (mc)
- * Un editor de textos (mcedit)

En el Shell tenemos algunas abreviaturas de tecla, entre las mas usadas están:

[INSERT]	Selecciona o desmarca un archivo o directorio
[CTRL]+[t]	Misma función que [INSERT]
[ALT]+[t]	Amplia o reduce el modo de ver el listado en pantalla
[CTRL]+[u]	Realiza un "swich" entre ambos paneles.
[ESC]+[+]	Permite marcar una serie de archivos según un filtro.
[ESC]+[-]	Permite desmarcar una serie de archivos según un filtro.
[CTRL]+[x] [s]	Realiza un enlace simbólico
[CTRL]+[x] [l]	Realiza un enlace "duro"
[CTRL]+[x] [c]	Chmod
[CTRL]+[x] [o]	Chown
[ENTER]	Ejecuta un archivo si este es "ejecutable"
[ALT]+[C]	Permite cambiar de directorio desde el panel
[CTRL]+[PAGE UP]	Realiza un "chdir" superior
[CTRL]+[PAGE DW]	Realiza un "chdir" inferior

En el editor:

[CTRL]+[r]	Permite grabar una macro
[CTRL]+[a]	Permite ejecutar una macro previamente grabada.

Parar y reactivar un servicio

killall -HUP <servicio>	Cierra todos los servicios abiertos asociados a un demonio en particular. (se reactiva con "<servicio> restart")
ejemplo	killall -HUP inetd killall -HUP httpd killall -HUP samba
	inetd restart httpd restart samba restart

Ejecutar periódicamente un comando - el demonio cron

El demonio "cron" es un proceso que arranca al encenderse la maquina y permanece activo por el tiempo mientras este encendida, ejecutando periódicamente ciertas ordenes.

El lapso mínimo que acepta es de 1 minuto.

El formato del archivo con que este demonio trabaja es especial, este archivo se encuentra en */etc/cron.d/kmod*

Para activar la cola en casos especiales "crontab ." (ver el punto al final)

minutos horas día_del_mes mes día_de_semana

Valores aceptados por el *cron*

minutos 0 a 59
horas 0 a 23
días del mes 0 a 31
meses 0 a 12
día de semana 0 a 6 (0 = domingo, 1 = lunes,...)

Ejemplos :

ejecutar algo 10 minutos después de las 02:00 am todos los días	10 2 * * * /home/fernando/bin/archivo
ejecutar algo a las 20:30 horas el 15 de cada mes	30 20 1 * * /etc/xxx
ejecutar algo cada 10 minutos	*/10 * * * * /etc/xxx
ejecutar algo cada 4 horas	* */4 * * * /etc/xxx
ejecutar algo cada minuto	* * * * * /etc/xxx
ejecutar algo desde las 8 AM hasta las 20 PM cada 2 horas	08-20/2 * * * * /etc/algo

Implementando servicios intranet / Internet

Básicamente utilizamos 2 servicios (aunque existen mas):

FTP y HTTP

Servicio FTP

Lo mas interesante de este servicio es que se activa en la instalación y es poco lo que requiere de nosotros, salvo el mantenimiento de sus sub-directorios (ubicados en */home/ftp*)

Restricciones y seguridad:

Toda la configuración sobre las diversas restricciones y la seguridad que se aplica a los accesos vía FTP esta en:

/etc/passwd
/etc/ftppaccess
/etc/ftpusers

passwd Archivo que contiene todas las cuentas del sistema. Aquí se debe incluir una cuenta ftp con restricción:
ftp::100:101:./home/ftp:/bin/false*

En la línea anterior le hemos indicado al sistema que el usuario *ftp* no pueda ejecutar el *shell* del sistema, en su lugar indicamos un *shell* que no

le permite acceder (false).

ftpusers Este archivo contiene los nombres de los usuarios que están impedidos de ingresar al sistema por FTP. Para reforzar la seguridad del sistema debemos incluir a *root* (en algunas distribuciones se le incluye por defecto) y a todo aquel usuario que no deseemos que tenga acceso por ftp al sistema.

ftpassess Este es el mas completo, se incluye la cuenta de correo del administrador del FTP (email), la ubicación del mensaje de bienvenida (mesagge). Tal vez su función mas importante sea permitir indicar que acciones están permitidas y cuales no, además de los directorios a los que podemos acceder, copiar y subir archivos. (véase ejemplo de archivo "ftpassess").

Como medida adicional de seguridad, se recomienda utilizar "chmod 555" en todos los directorios donde el usuario solo debe "leer") .

Estas medidas están orientadas al usuario "anónimo", ya que el usuario registrado en la red. obtendrá acceso directamente a su carpeta personal.

En Red Hat Linux 6.0, muchos servicios vienen pre-configurados de manera adecuada.... otros no. Un ejemplo de ello es el servicio de FTP, donde cualquier usuario puede acceder a la estructura completa del computador y no solo al directorio "/home/ftp" (directorio por defecto del ftp).

Otro punto es que si un usuario registrado accede vía FTP, este ingresa en su directorio personal, lo que le permite "subir" de directorio y entrar al sistema.

La mejor política es permitir el acceso vía FTP exclusivamente a usuarios "guest".

De todas maneras, existe modo de restringir el acceso al usuario "guest" o "annonymus" a los directorios que nosotros deseemos y esto se logra insertando las siguientes líneas en el archivo "ftpassess" :

```
# ESTABLECEMOS DIRECTORIO RAIZ PARA CADA TIPO DE
#USUARIO
anonymous-root      /home/ftp
guest-root           /home/ftp
users-root           /home/ftp
# PREVEEMOS CUALQUIER ACCION COBRE LOS ARCHIVOS
rename no           anonymous    # permiso para renombrar
delete no           anonymous    # permiso para borrar?
overwrite no        anonymous    # permiso para sobrescribir?
chmod no            anonymous    # permiso para cambiar permisos?
umask no            anonymous
# ARCHIVOS QUE NADIE PUEDE BAJAR
noretrive /etc/passwd /etc/group
noretrive core
# NUMERO MAXIMO DE INTENTOS PERMITIDOS DE LOGIN
loginfails 4
```

Servicio HTTP

Al igual que su primo FTP, este servicio se activa de manera automática. Lo único que debemos hacer para tener una pagina web es reemplazar el contenido de */home/httpd/html* por lo que queramos sea nuestra página web.

El archivo de configuración de este servicio se encuentra en :

`/etc/http/conf/httpd.conf`

Solo recuerda algo, añádele siempre que no indiques una pagina web un "/" al final. Linux es, para fines prácticos, un servidor Unix y distingue entre minúsculas y mayúsculas.

Si tu tienes un directorio llamado "mipagina", debes llamarlo así

`www.miweb.com/mipagina/` y no `www.miweb.com/mipagina`

En la primera, busca un directorio "mipagina", en la segunda, un archivo.....

En ambos casos, se tiene que ser "root" para copiar archivos a estos directorios, por lo que se debe designar a un usuario o grupo para manejar estos servicios (en caso de no ser nosotros) y cambiarle los permisos.

Ej. `chown .grupoX httpd -R` (ubicados en /home)

Correo electrónico interno

Cada usuario que ha sido creado en el servidor (o una maquina Linux), tiene por defecto una cuenta de correo interna con su nombre de usuario:

`<nombre_usuario>@<maquina>.<dominio>`
ejemplo.

`fernando@servidor.principal`

esto significa que el usuario "fernando", tiene una cuenta de correo en la maquina "servidor", perteneciente al dominio "principal".

Para utilizar el correo desde el prompt, basta digitar "mail", si se desea un programa "ligeramente" mas amigable, se puede utilizar el "pine". En todo caso, el X Windows proporciona varias interfases amigables para el manejo del correo.

`.forward` Este archivo permite que todo el correo que llegue a la cuenta de correo de un usuario, sea redirigido a una segunda cuenta. El contenido del archivo es la cuenta de destino. Por ejemplo, su Ud. es "root" y a su vez "X", posee dos cuentas diferentes, solo basta crear el archivo ".forward" dentro de ~/ de alguna de estas cuentas y derivarla a la otra. Así se asegura que una sola cuenta lea el contenido de ambas.

Ejemplo de archivo ".forward"

fernando@servidor.principal

Creando una conexión a Internet

Esta se basa en tres archivos fundamentales ubicados en el directorio `/etc/ppp`

`options` Contiene información de nuestro modem, velocidad de conexión y cientos parámetros que permitirán o no realizar una conexión estable

`pap-secrets` Contiene nuestro nombre de usuario y contraseña de acceso a internet

isp otorgada por nuestro proveedor.
Básicamente contiene el número de teléfono del proveedor (ISP) y las cadenas de conexión que se utilizarán.

a) archivo "options"

```
connect /etc/ppp/isp
crtcts
modem
passive
name tu_usuario # reemplazar por usuario verdadero
noipdefault
debug
defaultroute
asynctest a0000
pap-timeout 0
/dev/ttyS3 # esta línea varía según tu configuración
mtu 1500
115200 # la velocidad que soporta tu modem
```

b) archivo "isp"

```
#!/bin/sh
/usr/sbin/chat -v "" atdt2110500 CONNECT ""
```

c) archivo pap-secrets

```
"tu_usuario" * "tu_contraseña"
```

Debido a la naturaleza de las propiedades de cada versión de núcleo, es posible que este ejemplo funcione solo en algunos sistemas y no en todos.

A continuación, algunos comandos interesantes relacionados con Intranet / Internet:

ab Analiza un url y nos indica el software que controla ese sitio web, el puerto con el que se conecta, el tamaño del documento o pagina web, velocidad de transferencia, etc. Ejemplo:
ab http://161.132.5.133/
ab http://localhost/
ab http://www.rcp.net.pe/
Arroja datos interesantes. Lamentablemente solo funciona con servidores HTTP y no con FTP.

ckconfig Chequea que no falte ningún componente necesario para el sistema FTP.

Utilizando programas de Windows 95 y NT

Yo revente en una carcajada cuando supe que en Linux podía correr de manera bastante modesta pero efectiva algunos programas de las conocidas plataformas Win95 y WinNT.

Por ejemplo, yo tengo activado el servicio de "netlogon" que puede controlar las políticas en las terminales Win95 y WinNT pero en mi computadora solo tengo el Linux instalado, ¿Que hago?...

La solución :utilizo el "wine" o "WINDows Emulador". La sintaxis es sencilla, solo colocamos el comando wine seguido del programa que deseamos ejecutar.

```
wine /home/dosprogs/audio/goldwave.exe
```

```
wine /home/dosprogs/audio/poledit.exe
```

Definitivamente estos programas no han sido diseñados para Linux por lo que no será nada raro que algunos no funcionen, otros buscaran el registro de Windows y no lo encontraran.

Muchas de las opciones del "wine" ayudan a manejar mejor el programa, por ejemplo si deseamos emular un entorno "Windows NT", añadimos en parámetro

```
- winver win95|win31|nt40
```

Es recomendable conseguirse programas no tan modernos que no trabajen tan activamente con el registro de Windows 95/98/NT.

Por ejemplo, en este momento estoy utilizando mi programa FTP favorito, el CuteFTP para Windows 95 en Linux. Lo mejor de todo, Funciona!!! ¿los parámetros?...

```
wine -winver win95 /home/dosprogs/CuteFTP/cutftp32.exe
```

No se ve igual que en Windows, pero rinde igual que es lo importante.

Si desean, para no mantener una ventana "term" abierta, utilicen el parámetro & al final de la orden.

El archivo de configuración del "wine" es el "/etc/wine.conf", acá encontraran los parámetros y valores con los que trabaja este programa.

Algunos trucos y teclas escondidas

Es bastante raro que Linux se nos "cuelgue" en su modo nativo (modo texto), pero es normal que de cuando en cuando se cuelgue en modo Windows. Si se presenta el caso, "Nunca apagar sin que se haya descargado el sistema correctamente, sino utilizar [CTRL]+[ALT]+[BACKSPACE]".

[CTRL]+[ALT]+[BACKSPACE]
[CTRL]+[ALT]+[DEL]

Descarga el modo Xwindows
Archiconocida combinación de teclas, resetea la maquina. Cumple una función similar a usar *shutdown -r now*

[CTRL]+[D]

Fin de archivo (EOF). Muchos programas la utilizan para señalar el final del archivo o de una operación. Ojo, que [CTRL]+[D] desde el prompt sin ninguna utilidad cargada significa *logout*.

[CTRL]+[C]

Matar un proceso.

[CTRL]+[Z]

Detiene un proceso (no lo elimina).

[CTRL]+[H]

Simula la función del *backspace*

yes _order >/dev/null

Crea un proceso que satura la maquina si se repite unas 6 veces. Comando para probar la estabilidad del sistema.

información sobre Ficheros

Cada fichero tiene asociadas tres fechas:

- 1) Creación del fichero
- 2) Ultima modificación
- 3) Ultimo acceso

Editar nuestro perfil bash

Estos valores se pueden alterar con "touch".

Una de las cosas funciones mas importantes que el Unix ofrece a sus usuarios es la de "personalizar su shell".

El shell esta compuesto por varios archivos, los cuales almacenan información sobre nuestros gustos y manías. Lamentablemente no es tan automático como se desearía, se debe ingresar manualmente estos cambios.

Básicamente, el cambio que realicemos en uno es el que podrá ayudarnos a personalizar nuestro "shell", este archivo se llama ".bashrc" y esta ubicado en ~/

La sección que nos interesa es la de "alias", por ejemplo, podemos ingresar la línea "alias ll='ls -l'" (sin las doble comilla), desde ahora, cada vez que digitemos "ll" realmente se ejecutara "ll -l".

Ejemplos :

```
alias x='mc'  
alias h='cd ~/'  
alias ll='ls -l'
```

Variables de entorno

En el archivo ".bash_profile" encontraremos datos interesantes, además de permitirnos crear nuestras propias variables de entorno. Una de las primeras es USERNAME="XXX", cuando estemos en el prompt, digitemos echo \$USERNAME y se mostrara XXX. Esto nos puede ser muy útil se vamos ha realizar nuestros propios programas en Unix o Linux.

El símbolo \$ evalúa la variable de entorno, siempre y cuando nos encontremos en un shell .

Con el comando "env | more" (enviroment), podremos ver todas las variables definidas en nuestro entorno.

CAPITULO III

CONFIGURANDO Y PERSONALIZANDO LINUX

CAPITULO III

CONFIGURANDO Y PERSONALIZANDO LINUX

Ficheros de configuración de sistema

Todos los ficheros aquí comentados se encuentran dentro del directorio /etc

<i>/etc/motd</i>	Contiene el mensaje de bienvenida al sistema bien uno ingresa al sistema localmente.
<i>/etc/inetd</i>	
<i>/etc/smb.conf</i>	Contiene toda la información de los recursos compartidos de nuestra red.
<i>/etc/sysconfig/network</i>	Contienen el nombre de la maquina, este archivo debe ser editado en caso de que cambiemos de nombre a la maquina, después de utilizar "hostname <nuevo_nombre>".
<i>/etc/httpd/conf/httpd.conf</i>	Contiene directivas de control de operación del <i>server daemon httpd</i>
<i>/etc/httpd/conf/access.conf</i>	Contiene directivas del control de acceso a los documentos que el servidor proporciona a los clientes.
<i>/etc/ftpaccess</i>	Este archivo tiene toda la configuración del demonio FTP

Ficheros de configuración personal

Generalmente ubicados en ~/ (\$HOME)

Nota : Estos archivos varían de nombre según el sistema de ventanas que se este utilizando y si es que el usuario ha iniciado el entorno X.

<i>.kderc</i>	Script que el X Windows lee cada vez que arranca.
<i>.xinitrc</i>	Script que el X Windows lee cada vez que arranca.
<i>.twmrc</i>	Leído por el gestor Windows, twm.
<i>.fvwmrc</i>	Leído por el gestor Windows, fvwm.
<i>.emacs</i>	Leído por el editor "emacs" cada vez que arranca.
<i>.rhosts</i>	Permite usar de manera remota los programas de la maquina.
<i>.bashrc</i>	Contiene los alias y las funciones de cada usuario
<i>.bash_profile</i>	Contiene las configuraciones del entorno y los programas de inicio de cada usuario.
<i>.bash_logout</i>	Contiene las acciones que deben realizarse al cerrar la cuenta (hacer logout)
<i>.inputrc</i>	Contiene las combinaciones de teclas y otras informaciones.
<i>.message</i>	Es el mensaje inicial que se muestra en una conexión FTP. Puede estar ubicado en cualquier directorio donde exista algún acceso FTP.

Samba - comandos, trucos y configuración

Samba es un protocolo que permite la comunicación entre computadoras con diversos sistemas operativos, como Windows 9x, Windows NT, OS/2, AmigaOS, etc. con una computadora con Linux (sea servidor o cliente)

El archivo "principal" del samba es el "smb.conf", generalmente ubicado en /etc.

Es en este archivo donde podremos realizar las múltiples configuraciones que requiere para estar operativo.

Los comandos de samba son :

<i>samba start/stop</i>	Activa o detiene el servicio de samba
<i>smbclient</i>	Nos permite realizar múltiples operaciones, desde verificar los recursos compartidos de un host en particular (<i>smbclient -L <host></i>), hasta ingresar a un computador remoto de la siguiente manera : <i>smbclient \\\\<cliente>\<recurso> <password></i> si la conexión tiene éxito, nos saldrá este prompt smb: \>
<i>smbstatus</i>	Nota : Con este comando se pueden eliminar archivos ocultos, pero no los protegidos con el atributo de "solo lectura". Muestra el estado actual de las conexiones samba. Con esto se puede visualizar a los usuarios que están utilizando recursos compartidos de nuestro host.
<i>smbmount</i>	Permite montar un recurso compartido por red en una máquina con Linux. Ejemplo (montar el recurso compartido \\auxiliar\share en //mnt/share. <i>smbmount \\auxiliar\share /mnt/share -N</i>
<i>smbumount</i>	Desmonta una unidad montada con "smbmount".
<i>nmblookup <host></i>	Muestra el IP que esta siendo usado por el host indicado.
<i>testparm</i>	Revisa la configuración del archivo "smb.conf" y sugiere soluciones
<i>testprns</i>	Testea las impresoras
<i>smbpasswd</i>	Cambia la contraseña encriptada del samba y Windows NT
<i>smbadduser</i>	Añade un usuario a samba

Para manejar samba hay que conocer su archivo de configuración, el "smb.conf" esta conformado de la siguiente manera.

[global]	Aquí se indican todas las opciones "globales" del sistema
<i>workgroup = <grupo></i>	Grupo de trabajo al cual pertenece el host
<i>comment = <comentario></i>	Este comentario es el que se visualiza en las otras PC's
<i>mangle case = yes no</i>	Valor por defecto, "no". Ni idea, por defecto es "no"
<i>case sensitive = yes no</i>	Valor por defecto, "no". Indica si samba debe atender las diferencias entre minúsculas y mayúsculas.
<i>default case = upper lower</i>	Valor por defecto, lower
<i>preserve case =yes no</i>	Valor por defecto, yes. Preserva el nombre original como se escribió.
<i>short preserve case =yes no</i>	Valor por defecto, yes. Para sintaxis de nombres 8x3
[homes]	Se especifica la ubicación del directorio que contendrá los subdirectorios personales de los usuarios (normalmente en /home). Ejemplo :
	<i>[homes]</i> <i>path = /home/%S</i> <i>writeable = yes</i>
[printers]	Especifica la ubicación del "spool" de red. ejemplo :
	<i>[printers]</i> <i>path = /usr/spool/public</i> <i>writeable = no</i> <i>guest ok = yes</i> <i>printable = yes</i>

printcap name= lpstat	Obtiene automáticamente los nombres de la lista de impresoras del sistema.
[recurso_compartido]	Especifica un recurso compartido en el sistema
path = <ruta_recurso>	Directorio valido de la ubicación del recurso
read only = yes no	Lo activa como solo lectura o no
printable = yes no	Si el cliente puede abrir o no el servicio de impresión
guest ok = yes no	Indica si este recurso compartido en particular permite invitados
browseable = yes no	Si es que el recurso compartido puede ser explorable desde otro computador

Dentro de este archivo se pueden incluir un grupo de variables reemplazables, estas variables son reemplazadas por un valor específico :

%S	El nombre del servicio actual
%P	El directorio principal del servicio actual
%u	Nombre de usuario del servicio actual
%H	Directorio personal de %u
%g	Grupo primario de %u
%U	Nombre de la sesión activa
%G	Grupo primario de %U
%v	Versión de samba
%h	El hostname de internet donde samba corre
%m	El nombre NetBIOS de la maquina cliente
%L	El nombre NetBIOS del servidor
%M	El nombre internet de la maquina cliente
%N	Nombre del directorio raíz del NIS
%p	El la ruta del servicio obtenido por %N (%N:%p)
%R	El nivel de protocolo seleccionado después de la negociación
%d	El id del proceso en el servidor
%a	La arquitectura de la maquina remota
%l	El ip de la maquina cliente
%T	La fecha actual y la hora

Significado de las extensiones de los archivos

A diferencia del conocido Windows, donde casi todo archivo lleva una extensión por la que es reconocido, en Linux las extensiones muchas veces no existen y es difícil reconocer a un script de un archivo de información. Aunque básicamente están escritos en el mismo formato, su función es distinta. La mayoría de veces solo se les puede diferenciar por el directorio donde se encuentran.

.xpm	Iconos de aplicaciones (/usr/share/icons)
.conf	Archivos de configuración del sistema (ubicados por lo general en /etc)
.gz	Archivos comprimidos
.rpm	Archivos instaladores de programas (Red Hat)
.backup	Archivo de respaldo
.rc	Ficheros "run control", configuración

Variables internas y sus significados

\$LOGNAME	Nombre de "registro" del usuario que inicio sesión
-----------	--

<code>\$USERNAME</code>	Nombre completo del usuario <code>\$LOGNAME</code>
<code>\$USER</code>	Devuelve nuestro nombre de usuario (básicamente es igual a <code>\$USERNAME</code>)
<code>\$HOME</code>	Directorio personal del usuario actual
<code>\$PATH</code>	Ruta de localización de archivos de entorno
<code>\$HOSTNAME</code>	Devuelve el nombre del computador en el cual nos encontramos
<code>\$PWD</code>	Devuelve el nombre del directorio que contiene los ejecutables públicos
<code>\$ENV</code>	Devuelve el nombre del archivo que controla y contiene las variables de nuestro entorno
<code>\$BASH_ENV</code>	Devuelve el nombre del archivo que almacena la información del entorno bash del usuario
<code>\$SHELL</code>	Devuelve el nombre y ubicación del shell activo
<code>\$SHELLOPTS</code>	Devuelve las opciones del shell cargadas al inicio
<code>\$MAIL</code>	Devuelve la ubicación de la carpeta de correo personal
<code>\$HISTFILESIZE</code>	Máximo tamaño del file histórico de comandos
<code>\$HISTSIZ</code>	Tamaño actual del file histórico de comandos

Crear y eliminar una variable global (ejemplo)

```
# MINOMBRE="Fernando Fauche G." (Genera un valor de cadena)
# export MINOMBRE (convierte la cadena en variable global)
# echo $MINOMBRE (orden para mostrar el contenido de var)
# Fernando Fauche G. (resultado)
# set (para ver todas las variables)
# unset MINOMBRE (elimina la variable)
```

Posibilidades de las variables

A diferencia de otros lenguajes y sistemas operativos, Unix requiere anteponer una opción a la condición para darle un sentido. Así tenemos :

<code>-eq</code>	es igual
<code>-ne</code>	distinto de
<code>-gr</code>	mayor o igual
<code>-le</code>	menor o igual
<code>-lt</code>	menor que
<code>-gt</code>	mayor que
<code>[if -a <fichero>]</code>	existe <fichero>
<code>[if -f <fichero>]</code>	existe <fichero> y es archivo regular
<code>[if -g <fichero>]</code>	existe <fichero> y es set-group-id
<code>[if -h <fichero>]</code>	existe <fichero> y es un "link simbólico"
<code>[if -r <fichero>]</code>	existe <fichero> y es leíble
<code>[if -w <fichero>]</code>	existe <fichero> y es archivo con permiso de escritura
<code>[if -x <fichero>]</code>	existe <fichero> y es ejecutable
<code>[if -d <fichero>]</code>	existe <fichero> y es un directorio
<code>[if -s <fichero>]</code>	existe <fichero> y tiene tamaño mayor que 0
<code>[if -O <fichero>]</code>	existe <fichero> y es del actual usuario
<code>[if -G <fichero>]</code>	existe <fichero> y es del actual grupo

(para mayor referencia de estos parámetros, ejecutar "man bash")

Variables

<code>\$0</code>	El contenido es el nombre del script que estamos ejecutando
<code>\$1</code>	Primer parámetro del script
<code>\$2</code>	Segundo parámetro del script

\$#	Numero de parámetros que se han pasado al shell
\$*	Un argumento que contiene todos los parámetros menos el \$0
\$?	Numero donde se almacena el código de error del ultimo parámetro ejecutado
\$\$	Numero de proceso actual (PID)
\$_	Ultimo numero de proceso ejecutado

FUNCIONES

a) Creando condicionales con if

La siguiente es una estructura de una condicional compleja (o múltiple)

```
if <condicion>
then
  ...comandos...
else
  ...comandos...
else
  ...comandos...
else
  ...comandos...
fi
```

también puede hacerse con corchetes, este modo es el mas difundido. Nótese que existe un espacio entre los corchetes y los valores. Este espacio es necesario para que el sistema entienda correctamente lo que deseamos decir.

```
if [ <variable> = <valor> ]
then
  ...comandos...
else
  ...comandos...
else
  ...comandos...
else
  ...comandos...
fi
```

b) Creando secuencias con case

```
case <variable> in
then
  ...comandos...
else
  ...comandos...
else
  ...comandos...
else
  ...comandos...
fi
```

c) Creando secuencias con for

```
for j in *      (el shell cambia el * por los nombres de los archivos)
do
echo j
```

done

d) Creando secuencias con while

```
while <condición>  
do  
...comandos...  
done
```

e) Caracteres especiales para programación de scripts

<i>Backslash (\)</i>	Le quita todo significado especial a continuación de la barra invertida
<i>Comillas simples (')</i>	Le quitan todo significado especial a lo encerrado en "
<i>Comillas dobles (")</i>	Quitan todo significado especial a todos los caracteres con excepción de "\$", "\", y ""
<i>Com.de ejecucion (`)</i>	Ejecutan el comando encerrado entre las mismas y substituyen el valor por la salida estándar del comando ejecutado.

CAPITULO IV

EL NUCLEO DE LINUX

CAPITULO IV

EL NUCLEO DE LINUX

El código fuente de Linux

El directorio principal del código fuente es `"/usr/src/linux"`

Antes de continuar, es importante darle una leída al archivo "readme" ubicado en el directorio anteriormente mencionado.

Linux incluye el código fuente de su núcleo, con el cual podemos trabajar para modificarlo según nuestras necesidades.

Aquí la división de este directorio y su significado:

<i>arch</i>	Contiene todo el código específico de una arquitectura, dentro de este existen mas subdirectorios, uno por cada arquitectura soportada (ej. alpha, i386)
<i>include</i>	Contiene todos los ficheros necesarios para la compilación del núcleo. También puede obtener varios subdirectorios, uno por cada arquitectura soportada. Para cambiar de arquitectura hay que editar el fichero "Makefile" del núcleo y volver a ejecutar el programa de configuración del núcleo.
<i>init</i>	Este directorio contiene el código de iniciación del núcleo y es un buen punto de partida para comprender como funciona Linux.
<i>mm</i>	Aquí esta todo el código de gestión de memoria, independientemente de la arquitectura
<i>drivers</i>	Todos los manejadores de dispositivos se encuentran aquí. A su vez se divide en clases de controladores, como block para los dispositivos en bloques.
<i>ipc</i>	Este directorio tiene todo el código requerido para la comunicación entre procesos
<i>modules</i>	Este directorio se utiliza para montar los módulos cuando sea necesario
<i>fs</i>	Contiene todo el código del sistema de ficheros, a su vez se divide en subdirectorios, uno para cada tipo de ficheros soportado (vfat, ext2)
<i>kernel</i>	Aquí esta el código principal del núcleo. Una vez mas, aquella parte específica para una arquitectura se encontrará en "arch/*/lib/"
<i>net</i>	Código para trabajo en redes
<i>scripts</i>	Directorio donde se encuentran los scripts (tipo awk o tk) que son necesarios para configurar o compilar el núcleo.

Dónde empezar a mirar

"Cuando nos enfrentamos a consultar el código de un programa tan grande como el núcleo de Linux, en general no sabremos por dónde empezar. Todo se muestra como una gran cadena de la que no se ve su principio o final. Muchas veces comenzamos a estudiar el código y mirando aquí y

allá acabamos olvidando qué estábamos buscando. En los siguientes párrafos le daremos unas ideas sobre los puntos del código fuente donde puede encontrar respuestas a sus dudas."

Arranque e inicialización del sistema

"En un sistema basado en Intel, el núcleo comienza a ejecutarse cuando lo carga y le pasa el control un programa como loadlin.exe o LILO. Esta parte puede verse en arch/i386/kernel/head.S. Este programa inicial realiza ciertas preparaciones propias de la arquitectura y a continuación salta a la rutina main() del fichero init/main.c. "

Gestión de la Memoria

"Esta parte se encuentra principalmente en mm pero la parte más específica de cada arquitectura la encontraremos en arch/*/mm. El código de tratamiento de fallo de página se encuentra en mm/memory.c, y la parte correspondiente al mapeado de la memoria y la cache de páginas se encuentra en mm/filemap.c. La cache de buffer se detalla en mm/buffer.c y la cache del intercambio (swap), en mm/swap_state.c y mm/swapfile.c. "

El núcleo

"La parte común a todas las arquitecturas está en kernel, y la parte específica de cada una de ellas, en arch/*/kernel. El planificador lo encontramos en kernel/sched.c y el código para creación de procesos, en kernel/fork.c. La parte de bajo nivel de los manejadores se encuentra en include/linux/interrupt.h. La estructura de datos task_struct se localiza en include/linux/sched.h. "

PCI

"El pseudo-controlador de PCI se encuentra en drivers/pci/pci.c, con las definiciones especificadas en include/linux/pci.h. Cada arquitectura tiene una parte específica al correspondiente BIOS de PCI: los Alpha AXP se tratan en arch/alpha/kernel/bios32.c. "

Comunicación entre procesos

"Todo ello está en ipc. Los objetos del IPC de Unix Sistema V tienen una estructura ipc_perm y puede encontrarse en include/linux/ipc.h. Los mensajes de Sistema V se implementan en ipc/msg.c, la memoria compartida en ipc/shm.c y los semáforos en ipc/sem.c. Las tuberías se implementan en ipc/pipe.c. "

Tratamiento de interrupciones

"El código de tratamiento de interrupciones es en su mayor parte propio de cada microprocesador (y prácticamente distinto en cada plataforma). El código correspondiente a Intel (en un PC) está en arch/i386/kernel/irq.c y las definiciones necesarias se declaran en include/asm-i386/irq.h. "

Controladores de dispositivo

El grueso del código fuente de Linux lo forman los controladores de dispositivos. Todos ellos se encuentran bajo el directorio drivers, a su vez organizados según su tipo, en otros subdirectorios:

<i>/block</i>	Aquí están los dispositivos de bloque, como los discos IDE (en ide.c). Si queremos ver cómo los dispositivos pueden contener sistemas de ficheros e inicializarse, miraremos en la función device_setup() de drivers/block/genhd.c. Aquí, no solo se preparan discos: también la red si es necesario (por ejemplo, cuando queremos montar sistemas de ficheros nfs). Los dispositivos de bloques incluyen a los discos IDE y SCSI.
<i>/char</i>	En este directorio se pueden encontrar los dispositivos de carácter tales como los ttys, los puertos serie o el ratón
<i>/cdrom</i>	Aquí se encuentra todo el código referente a CDROMs especiales (como la interfaz con CD de SoundBlaster). Nótese que el

	controlador del CD tipo IDE/ATAPI se encuentra junto a los demás controladores IDE drivers/block/ide-cd.c) y que el encargado de los CD SCSI se encuentra en el fichero scsi.c de drivers/scsi
<i>/pci</i>	Aquí encontraremos el código fuente del pseudo-controlador de PCI. Es un buen sitio para ver cómo el subsistema PCI es mapeado e iniciado. El código específico para el PCI de Alpha AXP se encuentra en arch/alpha/kernel/bios32.c.
<i>/scsi</i>	Este es el lugar donde encontraremos todo lo referente al SCSI así como los diferentes controladores SCSI existentes y soportados en Linux
<i>/net</i>	En este directorio debemos mirar para buscar los controladores de tarjetas de red. Por ejemplo, el controlador de la tarjeta de red DECChip 21040 (PCI) se encuentra en tulip.c.
<i>/sound</i>	Aquí se implementa todo lo relacionado con las diferentes tarjetas de sonido.

Sistemas de Ficheros

Las fuentes para el sistema EXT2 están en el directorio fs/ext2/, residiendo las definiciones necesarias en include/linux/ext2_fs.h, ext2_fs_i.h y ext2_fs_sb.h. Las estructuras de datos correspondientes al Sistema de Ficheros Virtual (VFS) se declaran en include/linux/fs.h y el código está en fs/*. La implementación de la cache de buffer se reparte entre el archivo fs/buffer.c y el demonio update.

Redes

El código para el tema de las redes se encuentra en net y los ficheros de inclusión correspondientes en include/net principalmente. El código para los sockets BSD está en net/socket.c y el correspondiente a los sockets de IP versión 4, en net/ipv4/af_inet.c. El código de soporte de protocolo genérico (incluyendo las rutinas de manejo de sk_buff) está en net/core y la parte correspondiente a TCP/IP en net/ipv4. Los controladores de las tarjetas de red están en drivers/net.

Módulos

Para usar módulos, parte del código va incluido en el núcleo, y parte en el paquete de los módulos. El código del núcleo va todo él en kernel/modules.c, declarándose las estructuras y los mensajes del demonio kerneld en include/linux/module.h e include/linux/kerneld.h, respectivamente. Si desea conocer la estructura de un fichero objeto ELF, deberá mirar en include/linux/elf.h.

(conceptos extraídos de : El núcleo de Linux, escrita por David A Rusling)

Ordenes útiles en esta zona

Para desempaquetar todo el código fuente:

```
cd usr/src
gzip -cd linux-2.2.XX.tar.gz | tar xfv -
```

Para desempaquetar un parche

```
cd usr/src
gzip -cd patchXX.gz | patch -p0
```

ó

```
cd usr/src
bzip2 -dc patchXX.bz2 | patch -p0
```

Script que parcha de manera automática el núcleo:

```
cd /usr/src  
linux/scripts/patch-kernel
```

```
cd /usr/src/linux  
make mrproper
```

En este punto ya debe de tener los archivos fuentes correctamente instalados.

Para buscar una función dentro del kernel

```
nm vmlinux | sort | less
```

establecer una configuración apropiada para el núcleo

```
make config # tradicional
```

```
make menuconfig # por menus
```

```
make xconfig # muestra los menús en X
```

Después de alterar la configuración del kernel, ejecutar "make dep"

Secuencia de comandos para compilar el Kernel

- 1) make mrproper
- 2) make xconfig
- 3) make dep
- 4) make clean
- 5) make zimage (si el archivo resultante es un kernel muy grande intentar "make bzimage" y repetir desde el paso 2)
- 6) copiar el producto (zimage o bzimage) al directorio /boot
- 7) make modules
- 8) make modules_install
- 9) alterar lilo y actualizar

Nota para los usuarios de unidades Zip :

Se puede compilar todo o parte del sistema SCSI como módulos. Si hace esto, además de los comandos anteriores, deberá hacer make modules; make modules_install. Asegúrese de cargar scsi_mod.o, entonces sd_mod.o y finalmente el driver de su controladora, antes de acceder a la unidad ZIP SCSI

N de los T: Generalmente, de esto se encarga el sistema de forma transparente si previamente se han computado las dependencias entre módulos, (depmod -a al arrancar) al cargar módulos manualmente con el comando modprobe, o mejor aún, automáticamente si se usa kerneld .

Parchando o actualizando una versión de Linux

Cuando sale una versión mas actualizada del núcleo de nuestra distribución de Linux, no se hace necesario "reinstalar" para actualizar, simplemente ejecutar un comando para tal fin que incluye Linux.

```
cd /usr/src/linux
patch -p1 < patch-2.2.13-8.
```

el número anterior es un ejemplo, éste puede variar de acuerdo a la versión.

Controlando el arranque con LILO

Es importante mencionar que después de alterar en "algo" el archivo "lilo.conf", hay que ejecutar "lilo" para que valide y actualice esos cambios en el archivo de boot, caso contrario nos saldrá un error en el arranque y el sistema no levantará. (por lo que siempre es recomendable tener un disco hecho con la imagen "rescue")

Aquí los posibles errores que Lilo nos puede mostrar después de manipular su configuración.

Cuando LILO se inicia, si todo ha ido bien, presenta en la pantalla las cuatro letras "LILO". Como ya vimos, cada letra correspondía a una acción. Si LILO no ha funcionado, las letras que presente en pantalla junto con ciertos mensajes, serán la clave para saber donde está el problema. Veamos los más comunes:

"L" error:	El primer tramo del código ha sido cargado pero no el segundo. Esto se debe generalmente a un error físico en el sector de botado o a un problema de geometría del disco
"LI":	El segundo tramo del código ha sido invocado, pero no ha podido iniciarse. Esto sucede cuando hay un error de geometría de disco o se ha movido /boot/boot.b sin reinstalar LILO (/sbin/lilo)
"LIL":	El segundo tramo del código se ha iniciado pero no puede ubicar los datos que necesita en el fichero de mapeado. Esto puede deberse a un error físico del dispositivo de arranque, a un problema en la geometría del disco o un archivo boot no actualizado ubicado en otra partición (por ejemplo si utilizamos el Boot Loader de NT). En la mayoría de casos donde existe otro sistema de boot, se soluciona siguiendo los pasos que se describen en "Añadiendo el "boot" del Linux al Boot Loader del NT"
"LIL?":	El segundo tramo del código se ha cargado en una dirección de memoria equivocada. Esto se debe a un error en la geometría del disco o cuando se ha movido /boot/boot.b sin reinstalar LILO (/sbin/lilo)
"LIL-":	Los datos en el fichero de mapeado no son válidos. Las causas son las mismas que en el caso anterior

Añadiendo el "boot" del Linux al Boot Loader del NT

Uno de los dilemas mas frecuentes en el momento de instalar Linux es "de donde arranco si tengo ya un sistema operativo en mi PC". Las posibilidades de optar por el "lilo" son bastantes, pero también existen otras opciones, tal vez mas familiares que un programa totalmente extraño.

El siguiente ejemplo trata el "buteo" de Linux desde una PC con Windows NT previamente instalado.

Nota : Si se siguen las instrucciones de manera precisa, el sistema de arranque de NT no corre ningún riesgo.

1. Lo primero que tenemos que hacer es generar una imagen de 512 bytes del sector de arranque de Linux y plasmarla en un archivo. Esto se hace desde el Linux

```
dd if=/dev/hda8 of=/bootsect.lnx bs=512 count=1
```

en este ejemplo hemos creado el archivo "bootsect.lnx" que contiene una imagen de 512 bytes del sector de arranque de Linux. Esta imagen se copiará en "/"

Una vez obtenido este archivo, lo copiamos a la partición de arranque (que en DOS vendría a ser "C:\\" o /dev/hda1 en Linux)

2. Ya teniendo el archivo "bootsect.lnx" copiado en nuestra primera partición, procedemos a indicarle a NT que incluya al Linux como opción de arranque. Esta vez arrancamos en modo DOS y alteramos el archivo "C:\boot.ini" como se muestra a continuación. (es muy posible que este archivo tenga atributos de protección, en ese caso procedemos con el comando "attrib -h-r-s c:\boot.ini"

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINNT
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Windows NT Server"
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Windows NT Server(VGA Mode)"
C:\BOOTSECT.LNX="Linux"
```

3. Reiniciamos y escogemos.

Nota : Cualquier cambio en el boot del Linux o alteración en el "lilo.conf" debe ser seguido por la ejecución de "lilo" y la obtención de una nueva imagen (mas actualizada) del sistema, por lo que se deberá repetir el procedimiento en el punto "1)", no es necesario el punto "2)" para una actualización de boot.

Con otros sistemas, como Windows 95 / 98 o DOS, es necesario un programa que permita efectuar un "multiboot".

Debemos tener en cuenta la opción anteriormente mencionada es válida si Linux es el sistema operativo secundario o está instalado en una partición diferente a la primera y no se utiliza el "lilo" como sistema de arranque..

Configuración periféricos

Las siguientes configuraciones se añaden al archivo /etc/conf.modules

Placa de sonido incorporada (CMI8330)

```
alias sound ad1840
alias synth0 opl3
options ad1848 io=0x530 irq=7 dma=0 soundpro=1
options opl3 io=0x388
```

LECTURA RECOMENDADA

www.insflug.org	Un excelente lugar para encontrar "Cómos" en castellano.
www.lucas.es	El más grande centro de información de Linux en castellano. este es realmente un excelente punto de partida.
Linux Complete – Command reference	Un completo compendio de comandos y parámetros de Linux.
G.L.U.P.	Guía de Linux Para el Usuario
El núcleo del Linux	Escrita por David A Rusling
Guía de instalación oficial de Red Hat 5.0	Red Hat Software. Este manual también es conocido como GURH

Unix nunca fue diseñado para evitar que la gente haga cosas estúpidas, porque esa política les hubiera evitado hacer cosas inteligentes!!!! (Dowg Gwyn)
