

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)**

Formato para prácticas de laboratorio

PROGRAMA EDUCATIVO	PLAN DE ESTUDIO	CLAVE DE UNIDAD DE APRENDIZAJE	NOMBRE DE LA UNIDAD DE APRENDIZAJE
Ingeniero en Computación	2009-2	12124	Taller de Sistema Operativo Unix

PRÁCTICA No.	LABORATORIO DE	Taller de Sistema Operativo Unix	DURACIÓN (HORAS)
4	NOMBRE DE LA PRÁCTICA	Mandos para el manejo de archivos	2

1. INTRODUCCIÓN

Con frecuencia se describen los sistemas de Unix y Linux con el siguiente enunciado: "En un sistema Unix, todos es un archivo; si algo no es un archivo, es un proceso"¹. Se concluye entonces que los archivos y su manejo es muy importante en los sistemas Unix y Linux. En esta práctica se explicará cómo están organizados los archivos, cómo se realizan operaciones como copiar y borrar y también se explicará cómo se deben verificar las cuentas de usuario para evitar que sobrepase los límites impuestos por el administrador.

2. OBJETIVO (COMPETENCIA)

Utilizar los comandos básicos para el manejo de archivos con destreza y con una actitud de curiosidad por el funcionamiento interno de Linux.

3. FUNDAMENTO

Un **archivo** es un conjunto de datos que se guardan en un dispositivo y que tiene un nombre que lo identifica. Existen distintos tipos de archivos por ejemplo los archivos .mp3 que guardan música, los .jpeg que guardan imágenes y los .doc que guardan documentos.

Formuló Cecilia M. Curlango Rosas	Revisó Aglay González Pacheco	Aprobó	Autorizó David I. Rosas Almeida
Nombre y Firma del Maestro	Nombre y Firma del Responsable de Programa Educativo	Nombre y Firma del Responsable de Gestión de Calidad	Nombre y Firma del Director / Representante de la Dirección

Los archivos se guardan dentro de **directorios** que son grupos de archivos. Los directorios permiten organizar los archivos de tal forma que sea fácil encontrarlos cuando se necesiten.

En Linux existen dos tipos de directorios:

- **Directorio raíz.** Éste se representa con el símbolo `/` y es la raíz de todo el sistema de archivos. No puede borrarse ni renombrarse.
- **Subdirectorio.** Todos los directorios que se crean dentro de la raíz son subdirectorios. Éstos pueden borrarse y renombrarse. Cuando visualizamos el sistema de archivos en modo gráfico, los subdirectorios se representan como carpetas.

La siguiente figura muestra el contenido del directorio raíz del servidor **computacion.mx1.uabc.mx** el cual contiene tanto subdirectorios como archivos.



Algunos de estos subdirectorios son importantes para servidores de Linux. Es importante conocer qué función tienen los archivos que se encuentran en estos subdirectorios ya que esto facilita la administración de los equipos de cómputo con Linux. La siguiente tabla describe el contenido de estos subdirectorios.

Directorio	Descripción
<code>/</code>	Este es el directorio raíz. Todos los subdirectorios se guardan dentro de él. No se puede renombrar o borrar.
<code>/bin</code>	Este subdirectorio contiene muchos de los comandos que se ejecutan en Linux por ejemplo, <code>ls</code> , <code>cat</code> , <code>cd</code> .
<code>/etc</code>	Este subdirectorio contiene archivos de configuración del sistema y otros que también son importantes como el archivo de contraseñas password .
<code>/usr</code>	Este subdirectorio contiene otros subdirectorios en donde se guardan los programas que utilizan los usuarios y que pueden variar de un servidor a otro por ejemplo los editores como Kile , gedit y vi .

Directorio	Descripción
/home	Este subdirectorio contiene los subdirectorios en los que los usuarios guardan sus archivos. En el caso del servidor de computacion.mx1.uabc.mx los subdirectorios de los estudiantes y maestros están en el subdirectorio /externos/home y en el subdirectorio /home están los archivos de los usuarios administradores del servidor.
/sbin	Este subdirectorio contiene archivos de comandos que sólo puede ejecutar el superusuario del servidor o los usuarios con privilegios de administración. Uno de estos archivos es el comando reboot .

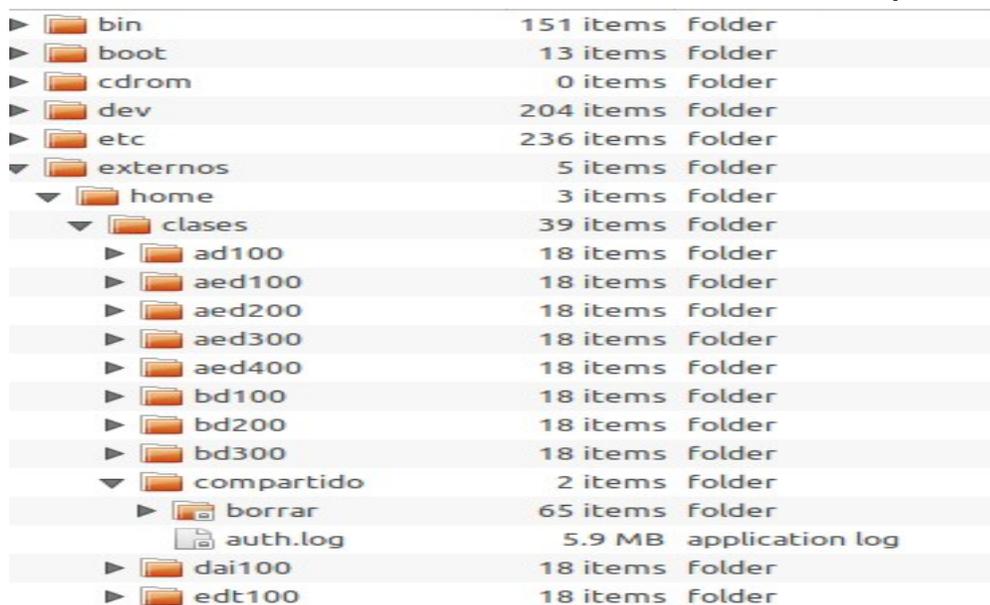
Rutas absolutas y rutas relativas

Existen dos formas de hacer referencia a un subdirectorio o archivo, con su ruta absoluta y con su ruta relativa. La **ruta absoluta** referencia a un subdirectorio sin importar en qué parte subdirectorio se encuentra actualmente el usuario. La ruta absoluta siempre empieza con el directorio raíz (*/*) e incluye todos los subdirectorios por los que se debe pasar para llegar al subdirectorio o archivo deseado.

Por el contrario, la **ruta relativa** depende de la ubicación actual del usuario. Las rutas relativas pueden hacer uso de los caracteres especiales descritos en la siguiente tabla:

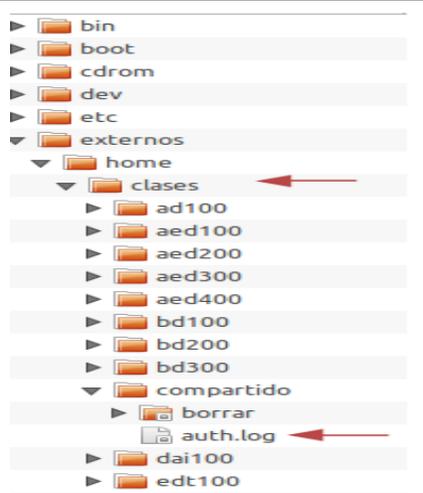
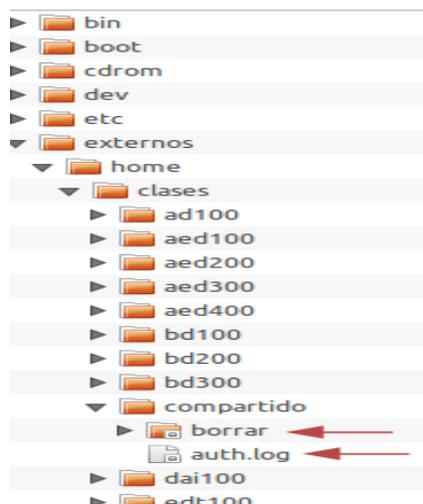
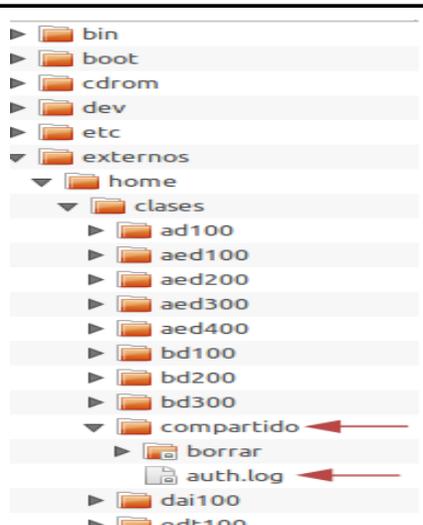
Caracter	Significado
.	El caracter punto representa al subdirectorio actual.
..	El caracter doble punto representa al subdirectorio inmediato anterior al actual.
~	El caracter tilde representa el subdirectorio del usuario.

La siguiente figura muestra el sistema de archivos del servidor. Ubique en la figura el archivo **auth.log**. La ruta absoluta de este archivo es: **/externos/home/clases/compartido/auth.log**.



La ruta relativa depende la ubicación actual del usuario. En la siguiente tabla se muestran ejemplos de rutas relativas. La primer columna muestra la ubicación actual del usuario en el sistema de archivos. La

segunda columna muestra la ruta relativa al subdirectorio de la primer columna para el archivo **auth.log**.

	Si el usuario está en:	La ruta relativa es:
	<code>/externos/home/clases</code>	<code>compartido/auth.log</code>
	<code>/externos/home/borrar</code>	<code>../auth.log</code>
	<code>/externos/home/clases/compartido</code>	<code>./auth.log</code>

	Si el usuario está en:	La ruta relativa es:
	/externos/home/clases/compartido	auth.log
	/etc	../externos/home/clases/compartido/auth.log

Comandos para recorrer subdirectorios

Para recorrer los subdirectorios del sistema de archivos de Linux desde una terminal se utiliza el comando **cd** cuyo nombre se deriva del inglés *change directory*.

Sintaxis

```
cd subdirectorio
```

En la siguiente figura se muestran algunos ejemplos del uso del comando de cambio de directorio. Inicialmente como indica el prompt el usuario se encuentra en su directorio de usuario `~`. Realiza un cambio a la raíz. El cambio nuevamente se puede verificar con el cambio en el prompt. Las siguientes instrucciones utilizan rutas absolutas. Después sube un nivel utilizando la ruta relativa `..` y posteriormente regresa al directorio de usuario.

```
curlango@computacion:~$ cd /
curlango@computacion:/$ cd /etc
curlango@computacion:/etc$ cd /externos/home/clases/compartido
curlango@computacion:/externos/home/clases/compartido$ cd ..
curlango@computacion:/externos/home/clases$ cd ~
curlango@computacion:~$ █
```

Es fácil perder la noción de dónde se encuentra dentro del sistema de archivos después de realizar varios cambios. Para conocer la ubicación actual del usuario en el sistema de archivos se utiliza el comando **pwd** cuya sintaxis se muestra a continuación.

Sintaxis

pwd

La siguiente figura muestra ejemplos de su uso.

```
curlango@computacion:/externos/home/clases$ pwd
/externos/home/clases
curlango@computacion:/externos/home/clases$ cd ..
curlango@computacion:/externos/home$ pwd
/externos/home
curlango@computacion:/externos/home$ cd .
curlango@computacion:/externos/home$ pwd
/externos/home
curlango@computacion:/externos/home$ cd ~
curlango@computacion:~$ pwd
/externos/home/maestros/_curlango
```

Comandos para manejar archivos

La manipulación de archivos se refiere a copiar, mover, borrar y mostrar su contenido. Estas operaciones y otras se acceden desde una terminal. Aunque también es posible realizarlas desde la interfaz gráfica, es importante conocer estos comandos ya que en algunos casos como en la administración de servidores, no se tiene acceso a una interfaz gráfica. Además las operaciones pueden realizarse con mayor agilidad desde la terminal.

En las descripciones de los comandos se utilizará la siguiente nomenclatura, cuando un argumento se muestra encerrado entre corchetes **[]**, indica que es opcional.

Listar archivos

Sintaxis

```
ls [-FlaRd][archivos]
```

La forma más sencilla de listar los archivos de un subdirectorio es utilizando únicamente el comando **ls** sin argumentos. En la siguiente figura se muestra un listado del contenido del directorio raíz. Compare el listado con el de la primera figura en esta práctica.

```
curlango@computacion:~$ cd /
curlango@computacion:/$ ls
bin  externos  lib64      opt  sbin  tmp      webmin-setup.out
boot home      lost+found proc  selinux usr
dev  initrd.img media      root  srv  var
etc  lib       mnt       run  sys  vmlinuz
```

En el listado anterior no se puede saber si algo es un directorio o un archivo. Con el argumento **-F** se agrega un símbolo al final de cada nombre para indicar de que tipo es por ejemplo, un **/** indica que se trata de un directorio, un ***** indica que es un programa y un **@** indica que es un enlace. En el primer ejemplo se muestra el contenido del directorio **raíz** y en el segundo un listado parcial del directorio **/bin**.

```
curlango@computacion:/$ ls -F
bin/  externos/  lib64/  opt/  sbin/  tmp/  webmin-setup.out
boot/ home/      lost+found/  proc/  selinux/  usr/
dev/  initrd.img@  media/  root/  srv/  var/
etc/  lib/       mnt/    run/  sys/  vmlinuz@
```

```
curlango@computacion:/$ ls -F bin
bash*      ip*          pidof@
bunzip2*  kbd_mode*   ping*
busybox*  kill*       ping6*
bzcat*    less*       plymouth*
bzcmp@    lessecho*   plymouth-upstart-bridge*
bzdifff*  lessfile@   ps*
```

Con el argumento **l** se mostrará en el listado información adicional acerca de cada archivo. La siguiente figura muestra un listado parcial del contenido del directorio **/etc**. La primera columna contiene los permisos del archivo. Estos permisos se interpretan de la siguiente forma. El primer símbolo, que en la figura aparece una **d** o un **-**. Con esto se indica el tipo de archivo, por ejemplo, **d** indica que es un subdirectorio y **-** indica que es un archivo. Los 9 símbolos restantes indican los permisos que tiene el archivo o subdirectorio. Se interpretan en grupos de 3. El primer grupo de 3 indica qué operaciones puede realizar el dueño con este archivo. El segundo grupo corresponde a las operaciones que tienen permitidas los miembros del grupo al que pertenece el archivo. Finalmente, el tercer grupo indica las operaciones que tiene permitidas el resto de los usuarios del servidor.

Las operaciones se indican con los símbolos **r w x y -**. El significado de los símbolos es:

- **r** => está permitida la lectura de este archivo
- **w** => está permitida la escritura en este archivo
- **x** => este archivo puede ejecutarse
- **-** => indica que no está permitida la operación por ejemplo, **r-x** no está permitida la escritura.

La segunda columna del listado de archivos indica la cantidad de enlaces que tiene este archivo. Un enlace es análogo a un *shortcut* en Windows.

La tercera indica el dueño del archivo y la cuarta columna indica el grupo al que pertenece el archivo.

La quinta columna es el tamaño del archivo. A ésta sigue la fecha y hora de la última modificación del archivo. Finalmente, se muestra el nombre del archivo.

```

curlango@computacion:/etc$ ls -l
total 1212
drwxr-xr-x  3 root root    4096 Jun 14 18:59 acpi
-rw-r--r--  1 root root    2981 Jun 14 18:36 adduser.conf
-rw-r--r--  1 root root     51 Jun 14 18:59 aliases
-rw-r--r--  1 root root   12288 Aug  3 01:38 aliases.db
drwxr-xr-x  2 root root    4096 Jun 21 11:07 alternatives
drwxr-xr-x  7 root root    4096 Jun 14 18:59 apache2
drwxr-xr-x  3 root root    4096 Jun 14 18:58 apm
drwxr-xr-x  3 root root    4096 Jul 26 04:40 apparmor
drwxr-xr-x  8 root root    4096 Jul 31 10:36 apparmor.d
drwxr-xr-x  4 root root    4096 Jul 26 04:42 apport
drwxr-xr-x  6 root root    4096 Jul 26 04:38 apt
-rw-r----- 1 root daemon  144 Oct 25 2011 at.deny
drwxr-xr-x  5 root root    4096 Jun 14 18:58 authbind
-rw-r--r--  1 root root    2076 Apr  3 08:58 bash.bashrc
-rw-r--r--  1 root root   58753 Mar 30 17:10 bash_completion

```

Cuando el nombre de un archivo inicia con un punto (.) entonces no aparece en el listado normal de archivos ya que estos archivos usualmente contienen configuraciones y no se necesitan modificar frecuentemente. Por esto el sistema los “esconde” de los listados de archivos para que no distraigan la atención de los demás archivos. Sin embargo, con el argumento **a** se incluyen estos archivos en el listado. La siguiente figura muestra un ejemplo de cómo mostrar los archivos escondidos. En la figura se comprueba que no sólo los archivos pueden esconderse, por ejemplo, **.bash_history** y **.bash_logout** sino que también los directorios pueden esconderse, por ejemplo, **.cache** y **.config**.

```

$ ls -al
total 248
drwxr-xr-x 29 curlango curlango 4096 Aug  7 11:29 .
drwxr-xr-x  3 root      root      4096 Aug  7 11:12 ..
-rw-----  1 curlango curlango 1320 Aug  7 11:26 .bash_history
-rw-r--r--  1 curlango curlango  220 Aug  6 12:56 .bash_logout
-rw-r--r--  1 curlango curlango 3486 Aug  6 12:56 .bashrc
drwx----- 16 curlango curlango 4096 Aug  6 16:43 .cache
drwx----- 13 curlango curlango 4096 Aug  7 11:26 .config
drwx-----  3 curlango curlango 4096 Aug  6 14:07 .dbus
drwxr-xr-x  2 curlango curlango 4096 Aug  6 14:07 Desktop
-rw-r--r--  1 curlango curlango   26 Aug  7 11:27 .dirc
drwxr-xr-x  2 curlango curlango 4096 Aug  6 14:07 Documents
drwxr-xr-x  2 curlango curlango 4096 Aug  7 11:27 Downloads

```

Otra opción del comando **ls** es **R** cuya función es mostrar no sólo el contenido del subdirectorio actual sino el contenido de los subdirectorios que están dentro del subdirectorio actual. La siguiente figura muestra un listado parcial en el que se incluye no sólo el contenido del subdirectorio **/usr** sino también el del subdirectorio **/usr/bin** que se encuentran dentro de él.

```
curlango@computacion:/usr$ ls -R
.:
bin
games
include
lib
local
lost+found
sbin
share
src

./bin:
[
2to3
2to3-2.7
a2p
ab
acpi_listen
addpart
addr2line
apport-bug
```

La opción **d** del comando **ls** muestra el nombre de los directorios en lugar de su contenido. En la siguiente figura se muestra el listado de todos los subdirectorios dentro de un subdirectorio.

```
$ ls * -d
Desktop      Downloads  examples.desktop  Pictures  Templates  workspace
Documents    Dropbox   Music              Public    Videos
```

La siguiente figura muestra el resultado que se obtendrá si no se utiliza la opción **d** del comando **ls**. Se muestra el nombre de los subdirectorios así como su contenido.

```
$ ls *
examples.desktop

Desktop:

Documents:

Downloads:
com.google.gdt.eclipse.suite.4.2.update.site_3.0.1.zip
dropbox_1.4.0_amd64.deb
eclipse-SDK-4.2-linux-gtk-x86_64.tar.gz
google-chrome-stable_current_amd64.deb
jdk-6u33-linux-x64.bin

Dropbox:
ArticuloJUCS      Compartidos      Getting Started.rtf  Mio      Stuff
articuloSalsa    CongresoMujer   iAnnotate PDF       phidgets  Tesis
Cecilia Curlango Datos            laFam                Photos    TWeb
Clases            fotos proyectos London2012            Public    UCSD_Stuff

Music:

Pictures:

Public:

Templates:

Videos:

workspace:
```

Listar el contenido de archivos

El comando **cat** se utiliza para visualizar en la pantalla el contenido de archivos. Este comando sólo se puede utilizar con archivos cuyo contenido es texto puro, esto significa que no se puede utilizar para visualizar el contenido de una imagen. Para utilizarlo, sólo se debe indicar el nombre del archivo o de los archivos que se quieren visualizar. En la sección de referencias de esta práctica se encuentra la liga al manual de este comando. Ahí se ve que **cat** tiene algunas opciones que se pueden utilizar para dar formato al listado del contenido que se muestra.

Sintaxis

```
cat [archivo(s)]
```

La siguiente figura muestra un ejemplo del uso de **cat**.

```
$ cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      curlango-VirtualBox

# The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0      ip6-localnet
ff00::0      ip6-mcastprefix
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```

El contenido de algunos archivos es tan extenso que no se puede visualizar su contenido usando **cat** en una sola pantalla. Para estos archivos se puede utilizar el comando **more** que muestra el contenido de archivos (o flujos) una pantalla a la vez. Esto significa que muestra todo el contenido que cabe en una pantalla y espera a que el usuario presione la **barra espaciadora** para mostrar la siguiente pantalla con contenido. También se utiliza la tecla **Enter** y para mostrar la siguiente línea de un archivo. La tecla **q** se utiliza para terminar de mostrar el archivo antes de llegar a su fin. En la sección de referencias se encuentra la liga al manual de **cat**.

Sintaxis

```
more [archivo(s)]
```

La siguiente figura muestra un ejemplo del uso de **more** con el archivo **/etc/password**.

```

$ more /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
--More-- (42%)

```

Es común utilizar `more` para mostrar por páginas el resultado de algún otro comando. Esto requiere el uso de los comandos de redireccionamiento de salida que se estudiaron anteriormente. El siguiente ejemplo muestra cómo se utiliza `more` conjuntamente con entubamiento para mostrar el listado de archivos en `/etc` una pantalla a la vez.

```

$ ls -l /etc | more
total 1500
drwxr-xr-x  3 root    root      4096 2011-12-09 11:01 acpi
-rw-r--r--  1 root    root      2986 2009-04-20 06:59 adduser.conf
-rw-r--r--  1 root    root        47 2012-05-04 12:42 adjtime
-rw-r--r--  1 root    root        51 2011-05-20 12:25 aliases
-rw-r--r--  1 root    root     12288 2011-05-20 12:25 aliases.db
drwxr-xr-x  2 root    root     12288 2012-08-01 09:40 alternatives
-rw-r--r--  1 root    root      395 2008-09-02 16:37 anacrontab
drwxr-xr-x  7 root    root      4096 2012-03-20 10:01 apache2
drwxr-xr-x  7 root    root      4096 2011-01-06 16:03 apm
drwxr-xr-x  3 root    root      4096 2012-06-13 09:42 apparmor
drwxr-xr-x  7 root    root      4096 2012-06-12 11:58 apparmor.d
drwxr-xr-x  4 root    root      4096 2012-02-17 09:25 appport
drwxr-xr-x  6 root    root      4096 2012-03-09 10:11 apt
-rw-r----- 1 root    daemon    144 2009-04-17 00:53 at.deny
drwxr-xr-x  3 root    root      4096 2011-03-07 13:52 avahi
-rw-r--r--  1 root    root      1939 2010-04-18 18:51 bash.bashrc
--More--

```

Administrar archivos

En esta sección se describe el funcionamiento de los comandos `cp`, `mv`, `rm` y `ln` que se utilizan en para administrar archivos. Administrar se refiere a aquellas acciones que se realizan para organizar el sistema de archivos, específicamente copiar un archivo, mover un archivo a otro sitio, borrar un archivo y crear un acceso rápido para el archivo.

Sintaxis

cp fuente destino

En su forma más sencilla el comando **cp** copia el archivo que se indica como *fuentes* al sitio indicado como *destino*. Cuando se hace una copia se termina la operación con dos ejemplares del archivo, el original y la copia. Como se muestra en la siguiente figura, se utilizan comodines para copiar más de un archivo. En el ejemplo se copian todos los archivos del subdirectorio **/etc** cuyo nombre empiece con **alias** y el destino es un subdirectorio llamado **./copiaEtc**. Después de la copia se comprueba la operación listando los archivos en **./copiaEtc**.

```
$ cp /etc/alias* copiaEtc
$ ls copiaEtc
aliases aliases.db
```

Sintaxis

mv fuente destino

En su forma más sencilla el comando **mv** mueve el archivo que se indica como *fuentes* al sitio indicado como *destino*. Cuando se hace el movimiento, se termina la operación con sólo un ejemplar del archivo, el original pero en un sitio diferente. Además de mover archivos, **mv** también se utiliza para cambiarle el nombre a un archivo. En el ejemplo de la siguiente figura, primero se mueve el archivo **alias** al subdirectorio **resguardo**. Se listan los subdirectorios para comprobar que **alias** ya está dentro del subdirectorio **resguardo**. Con la siguiente instrucción cambia el nombre del archivo **alias** al nombre **alias.alien** y se lista el subdirectorio para comprobar la operación. Se comprueba que ya no existe un archivo que se llame **alias.db**.

```
$ ls
aliases aliases.db resguardo
$ ls resguardo/
$ mv alias resguardo
$ ls
aliases.db resguardo
$ ls resguardo/
aliases
$ mv aliases.db aliases.alien
$ ls
aliases.alien resguardo
```

Sintaxis

rm [-irf] archivo(s)

En Unix y Linux cuando se borra un archivo desde la línea de comandos, no hay forma de recuperarlo, por esta razón cuando se utiliza el comando **rm** se debe tener cuidado de que en realidad se quiera borrar el archivo.

Para poder borrar un archivo del que no se es el dueño, se debe tener permiso de escritura sobre el archivo. Si el dueño entonces se puede borrar el archivo sin importar los permisos.

En su forma más simple, borrar un archivo sólo requiere especificar el nombre del archivo al comando **rm**.

Las tres opciones siguientes se utilizan con el comando `rm` para cambiar su comportamiento por default:

- `-i` => activa el modo interactivo, antes de borrar cada archivo pedirá que se confirme la operación.
- `-r` => activa el modo recursivo, entra a los subdirectorios y borra su contenido y después borra el subdirectorio en sí.
- `-f` => no muestra mensaje de error si no encuentra el archivo que se va a borrar.

La siguiente figura muestra algunos ejemplos del uso del comando `rm`.

```
$ ls -l
total 4
drwxr-xr-x 2 curlango curlango 4096 2012-08-08 18:00 resguardo
-rw-r--r-- 1 curlango curlango 0 2012-08-08 18:01 u1
-rw-r--r-- 1 curlango curlango 0 2012-08-08 18:01 u2
-rw-r--r-- 1 curlango curlango 0 2012-08-08 18:01 u3
-rw-r--r-- 1 curlango curlango 0 2012-08-08 18:01 u4
$ rm u5
rm: cannot remove `u5': No such file or directory
$ rm -f u5
$ rm -i u2 u3
rm: remove regular empty file `u2'? y
rm: remove regular empty file `u3'? y
$ ls
resguardo u1 u4
$ rm resguardo
rm: cannot remove `resguardo': Is a directory
$ rm -rf *
$ ls
$
```

Sintaxis

In fuente destino

Los enlaces son accesos directos a otros archivos. Son similares a los *shortcuts* en Windows. Es común encontrar en el sistema de archivos de Unix o Linux enlaces a otros archivos. Usar enlaces en lugar de hacer copias de archivo tiene las siguientes ventajas:

1. Ocupa menos espacio en disco un enlace que una copia de un archivo.
2. Si se modifica el archivo o cualquiera de los enlaces a él, los cambios se reflejan en todos los enlaces automáticamente.
3. Los enlaces pueden tener un nombre diferente al archivo original.

La siguiente figura muestra algunos ejemplos del uso de enlaces.

```
$ ln u2 resguardo/ligaU2
$ ls
resguardo u1 u2 u3 u4
$ ls resguardo/
ligaU2 u1
$ cat u2
Este es el contenido del archivo
u2. Solo tiene
tres lineas.
$ cat resguardo/ligaU2
Este es el contenido del archivo
u2. Solo tiene
tres lineas.
$ ln u2 resguardo
$ ls resguardo/
ligaU2 u1 u2
$ cat resguardo/u2
Este es el contenido del archivo
u2. Solo tiene
tres lineas.
```

Administrar el espacio en disco asignado

En el servidor de clases, los estudiantes tienen limitado el espacio en disco que pueden utilizar para sus archivos. Cuando se van acercando a este límite o lo sobrepasan automáticamente se les bloquea el acceso a sus cuentas en modo gráfico y solo pueden acceder desde modo texto. Es importante que los estudiantes conozcan los comandos que se utilizan para administrar el espacio en disco que ocupan en sus cuentas. En esta sección se presentará el comando **quota** que es útil para este propósito. Adicionalmente, se presentarán los comandos **du** y **df** con los cuales se puede conocer el estado actual del espacio en los discos.

Sintaxis

```
quota [-s]
```

El comando **quota** muestra la cantidad de espacio que tiene asignado un usuario y cuánto espacio tiene ocupado actualmente. Como se muestra en la siguiente figura, el espacio lo muestra por bloques y por archivos. Con la opción **s** muestra las cantidades en unidades que son más fáciles de entender para los usuarios por ejemplo, megabytes, kilobytes y gigabytes.

El listado que genera **quota** se interpreta de la siguiente forma, la columna con encabezado **quota** indica el tope de espacio asignado. Si se sobrepasa este valor, se ha excedido la cuota y el usuario tiene el tiempo indicado en la columna **grace** para borrar archivos y quedar por debajo del límite. La columna **limit** es el tope máximo; cuando se llega a este tope, el usuario ya no puede grabar mas archivos.

```
[aguilar@computacion ~]$ quota
```

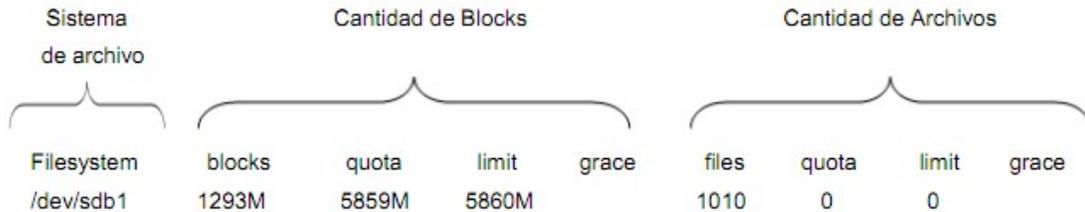
```
Disk quotas for user aguilar (uid 501):
```

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/sdb1	1323124	5999000	6000000		1010	0	0	

```
[aguilar@computacion ~]$ quota -s
```

```
Disk quotas for user aguilar (uid 501):
```

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/sdb1	1293M	5859M	5860M		1010	0	0	



Sintaxis

`df`

El comando **df** muestra un listado del espacio disponible en disco del servidor. La siguiente figura muestra un ejemplo de su uso en un servidor. La columna **Use%** indica que porcentaje de la partición está siendo utilizada.

```
$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/sda2      68242252 23762864 41061432 37% /
udev            1970988      4 1970984 1% /dev
tmpfs           791908      304 791604 1% /run
none             5120         0 5120 0% /run/lock
none            1979768      0 1979768 0% /run/shm
/dev/sda5       7786092    246028 7149512 4% /tmp
/dev/sda3      97503756 25070996 67549972 28% /externos
/dev/sda6      77997496 2340576 71750720 4% /usr
/dev/sda7      54701264 1435864 50525076 3% /var
```

Sintaxis

`du [-abch] archivo(s)`

El comando **du** muestra un listado del espacio ocupado en el disco del servidor. Las opciones de este comando permiten especificar el tipo de listado que se quiere. El significado de las opciones es:

- **a** => mostrar el espacio que ocupan todos los archivos no sólo los directorios
- **b** => mostrar en bytes el espacio ocupado
- **c** => mostrar el gran total
- **h** => mostrar los espacios ocupados en forma legible por los humanos, por ejemplo 1K, 2M, 5G.

La siguiente figura muestra un ejemplo del uso de **du** para mostrar el espacio en formato legible por humanos que ocupan los archivos y directorios que empiezan con **D**.

```
curlango@computacion:~$ du -h D*
560K   Desktop
1.8M   Documents
```

En las figuras mostradas a continuación se muestran los archivos y directorios que empiezan con **D** incluyendo archivos escondidos. También se ve el contraste entre usar o no la opción **h**.

<pre>curlango@computacion:~\$ du -a D* 4 Desktop/.directory 8 Desktop/Home2.desktop 72 Desktop/jspcheatsheet.pdf 468 Desktop/clase09.pdf 4 Desktop/trash.desktop 560 Desktop 1000 Documents/lshort.pdf 112 Documents/teachingLatexWhyHow.pdf 24 Documents/license.txt 12 Documents/latex.odt 4 Documents/.directory 92 Documents/kile2.jpeg 96 Documents/kile1.jpeg 48 Documents/kile3.jpeg 48 Documents/kile4.jpeg 56 Documents/practica5.pdf 316 Documents/clase05.pdf 1812 Documents</pre>	<pre>curlango@computacion:~\$ du -ha D* 4.0K Desktop/.directory 8.0K Desktop/Home2.desktop 72K Desktop/jspcheatsheet.pdf 468K Desktop/clase09.pdf 4.0K Desktop/trash.desktop 560K Desktop 1000K Documents/lshort.pdf 112K Documents/teachingLatexWhyHow.pdf 24K Documents/license.txt 12K Documents/latex.odt 4.0K Documents/.directory 92K Documents/kile2.jpeg 96K Documents/kile1.jpeg 48K Documents/kile3.jpeg 48K Documents/kile4.jpeg 56K Documents/practica5.pdf 316K Documents/clase05.pdf 1.8M Documents</pre>
---	--

4. PROCEDIMIENTO (DESCRIPCIÓN)

A) EQUIPO NECESARIO

MATERIAL DE APOYO

Computadoras con Linux instalado y atado al servidor.

B) DESARROLLO DE LA PRÁCTICA

1. Describe con tus propias palabras para qué se utilizan las siguientes opciones del comando **ls**.

Opción	Funcionamiento
F	
l	
a	

Opción	Funcionamiento
R	
d	

2. Muestra todos los archivos del subdirectorio **/etc** que empiecen con la letra **n** sin mostrar el contenido de los subdirectorios.
3. Copia a tu subdirectorio los archivos del subdirectorio **/etc** que empiezan con la letra **s** y tengan la terminación **.conf**.
4. Muestra por páginas el archivo **/etc/hdparm.conf**.
5. Crea una copia del archivo **.bash_history** poniéndole el nombre **historial**.
6. Crea un enlace llamado **enlace_historial** al archivo **historial**.
7. Utilizando un editor de texto inserta tu nombre al inicio del archivo **enlace_historial**. Verifica que el archivo **historial** ahora también tiene tu nombre.
8. Verifique el espacio que están ocupando sus archivos en el disco del servidor. ¿Cuánto espacio tiene disponible aún?
9. Verifique cuál de sus subdirectorios es el que está ocupando mas espacio en disco. ¿Cuánto espacio está ocupando?
10. Haga una copia del subdirectorio que está ocupando mas espacio en disco. Verifique nuevamente ¿cuánto espacio tiene disponible antes de exceder su cuota?
11. Cambiese al subdirectorio del nivel anterior tres veces. Muestre con un comando el nombre del directorio en el que se encuentra.
12. Intente mover todos los archivos que se encuentran en **/usr/bin** que empiezan con **mk**. Explique por qué tuvo éxito o no la operación.
13. ¿Qué porcentaje de **/home** está desocupado?
14. Consulta el manual del comando **cat** para encontrar la opción requerida para mostrar el contenido del archivo **historial** numerando cada línea.
15. Genera un listado largo de los archivos **historial** y **enlace_historial**. ¿Cuántos enlaces marca cada uno de estos archivos?
16. Crea otro enlace al archivo **historial** llamado **tercer_historial**. Verifica nuevamente el número de enlaces que marca cada uno.
17. Borra los archivos **enlace_historial** y **tercer_historial**. ¿Cuántos enlaces marca el archivo?

C) CÁLCULOS (SI APLICA) Y REPORTE

5. RESULTADOS Y CONCLUSIONES

6. ANEXOS

7. REFERENCIAS

1. Introduction to Linux, Chapter 3 About files and the file system. http://tldp.org/LDP/intro-linux/html/sect_03_01.html
2. Manual de **ls** <http://manpages.ubuntu.com/manpages/hardy/man1/ls.1.html>
3. Manual de **cat** <http://manpages.ubuntu.com/manpages/precise/man1/cat.1.html>
4. Manual de **more** <http://manpages.ubuntu.com/manpages/precise/man1/more.1.html>
5. Manual de **cp** <http://manpages.ubuntu.com/manpages/precise/man1/cp.1.html>
6. Manual de **mv** <http://manpages.ubuntu.com/manpages/precise/man1/mv.1.html>
7. Manual de **rm** <http://manpages.ubuntu.com/manpages/precise/man1/rm.1.html>
8. Manual de **ln** <http://manpages.ubuntu.com/manpages/precise/man1/ln.1.html>
9. Manual de **quota** <http://manpages.ubuntu.com/manpages/precise/man1/quota.1.html>
10. Manual de **df** <http://manpages.ubuntu.com/manpages/precise/man1/df.1.html>
11. Manual de **du** <http://manpages.ubuntu.com/manpages/precise/man1/du.1.html>