

Comandos básicos

Uno de los comandos más básicos que empezaremos a utilizar será el **ls** que nos permite poder ver los contenidos de los directorios.

Opciones :

-l	Muestra los archivos en forma de columna.
-a	Muestra los archivos que empiezan con . (ocultos).
-R	Listado recursivo de todos los archivos y subdirectorios.
-i	Muestra el número de inodo.
-h	Muestra el espacio ocupado del archivo en MB, KByte, etc.
-t	Ordena por la fecha y hora de modificación.

Ejemplos :

```
# ls -la

drwxr-xr-x 19 root root 4096 Feb 18 15:01 .
drwxr-xr-x 29 root root 4096 Feb 18 18:02 ..
drwx----- 2 root root 4096 Sep  6 2009 .aptitude
-rw----- 1 root root 7992 Feb 23 16:30 .bash_history
-rw-r--r-- 1 root root 412 Dec 15 2004 .bashrc

# ls /

bin    initrd.img  media      sbin  sys    vmlinuz
boot   dev        initrd.img.old  proc  selinux tmp    vmlinuz.old
cdrom  etc        lib        mnt   srv    usr
home  lost+found opt        root  var
```

Entrar a los directorios : **cd**

Para movernos en distintos directorio/subdirectorios usamos el comando **cd**.

Ejemplos :

```
# cd /bin

# cd /usr/share/doc

# cd /home/pablo
```

Para ir a un directorio anterior :

```
# cd ..
```

Nos permite ir a nuestro directorio *home* que es donde entramos al sistema.

```
# cd ~
```

Obtener información de los comandos

Gracias a una serie de comandos podemos obtener información de los archivos de configuración, y de los comandos que ejecutamos, también tendremos otros comandos que nos permite realizar búsquedas de manuales .

Comando : *apropos*

Cada manual contiene una pequeña descripción, este comando nos permite buscar dentro de esas descripciones.

Opciones :

-e, --exact	Cada palabra clave se comparará de forma exacta con los nombres de páginas y las descripciones.
-d, --debug	Imprime información de depuración.
-r, --regex	Interpreta cada palabra clave como una expresión regular. Este comportamiento es el predeterminado. Cada palabra clave se comparará con los nombres de las páginas y la descripciones.
-w, --wildcard	Idem a la opción -r, --regex salvo que permite utilizar comodines.
-M ruta, --manpath=ruta	Especifica rutas separadas por “dos puntos” de la ubicación de los manuales. Por defecto apropos utiliza la variable de entorno \$MANPATH en todo caso que este vacía la variable de entorno utiliza como ruta para los manuales la variable \$PATH .
-h, --help	Ayuda.

Ejemplos :

```
# apropos ls
```

```
# apropos -w ls
```

Comando : *info*

Este comando supera la información del **man**, el comando **info** nos permite navegar por medio de los enlaces como si fuera una página web.

Opciones :

-k, --aprops	Buscar cadena en todos los índices de todos los manuales.
-d, --directory=DIR	Agrega un directorio (DIR).
-f, --file=ARCHIVO	Se especifica un archivo determinado.
-M ruta, --manpath=ruta	Especifica rutas separadas por “dos puntos” de la ubicación de los manuales. Por defecto apropos utiliza la variable de entorno \$MANPATH en todo caso que este vacía la variable de entorno utiliza como ruta para los manuales la variable \$PATH .
-h, --help	Ayuda.

Ejemplos :

```
# info bc
# info -f /usr/share/info/bc.info.gz
```

Dentro del directorio **/usr/share/info** se encuentra los archivo de información comprimida.

Comando : *whatis*

Nos da una pequeña descripción de un comando y las secciones de **man** que podemos consultar.

Opciones :

-d, --debug	Imprime información de depuración.
--------------------	------------------------------------

-r, --regex	Interpreta cada palabra clave como una expresión regular. Este comportamiento es el predeterminado. Cada palabra clave se comparará con los nombres de las páginas y la descripciones.
-w, --wildcard	Idem a la opción -r, --regex salvo que permite utilizar comodines.
-M ruta, --manpath=ruta	Especifica rutas separadas por “dos puntos” de la ubicación de los manuales. Por defecto apropos utiliza la variable de entorno \$MANPATH en todo caso que este vacía la variable de entorno utiliza como ruta para los manuales la variable \$PATH .
-h, --help	Ayuda.

Ejemplos :

```
# whatis bc
# whatis -w ls*
```

Comando : **man**

Vemos los manuales de los programas y archivos de configuración.

La siguiente tabla muestra los números de sección del manual y los tipos de páginas que contienen.

1	Programas ejecutables y guiones del intérprete de órdenes.
2	Llamadas del sistema (funciones servidas por el núcleo).
3	Llamadas de la biblioteca (funciones contenidas en las bibliotecas del sistema).
4	Ficheros especiales (se encuentran generalmente en /dev).
5	Formato de ficheros y convenios p.ej. /etc/passwd .
6	Juegos.
7	Paquetes de macros y convenios p.ej. man(7) , groff(7) .
8	Órdenes de admistración del sistema (generalmente solo son para root).
9	Rutinas del núcleo [No es estándar].
n	Nuevo [obsoleto].
l	Local [obsoleto].
p	Público [obsoleto].
o	Viejo [obsoleto].

Opciones :

-d, --debug	Imprime información de depuración.
-t <i>ls</i>	Formatea la página del manual <i>ls</i> en postscript.
-k <i>ls</i>	Idem <i>apropos</i> .
-f <i>ls</i>	Busca las páginas del manual referenciadas por <i>ls</i> e imprime la descripción de las que encuentra. Equivalente al comando <i>whatis</i> .
-u, --update	Los caches de los índices de las bases de datos son actualizados sobre la marcha, es decir no es necesario que <i>mandb</i> se ejecute periódicamente para mantener la consistencia.
-M ruta, --manpath=ruta	Especifica rutas separadas por “dos puntos” de la ubicación de los manuales. Por defecto apropos utiliza la variable de entorno \$MANPATH en todo caso que este vacía la variable de entorno utiliza como ruta para los manuales la variable \$PATH .
-P pagina, --pager=pagina	La <i>pagina</i> es el tamaño, esta sentencia sobrescribe la variable de entorno \$PAGER .
-d, --debug	Imprime información de depuración.

Ejemplos :

```
# man ls
# man -k ls
# man -t ls | ps2pdf - > ls.pdf
```

Dentro del archivo */etc/manpath.config* contiene los directorios donde se busca la información de los archivos man.

Creación de un archivo vacío o cambio de fecha y hora: *touch*

Este comando nos sirve para dos motivos, es decir si no existe un archivo me lo crea con la fecha de hoy, y si existe le cambia la fecha y hora actual al archivo.

Opciones :

-a	<i>Cambia solamente el tiempo de acceso.</i>
-d --date=string	<i>Se le pasa una fecha como string.</i>
-m	<i>Solamente cambia hora.</i>

Ejemplos :

```
# touch mi_archivo_vacio

# ls -l mi_archivo_vacio

-rw-r--r-- 1 root root 0 Sep 28 10:25 mi_archivo_vacio
```

Copia de archivos: *cp*

El comando *cp* nos permite copiar un archivo o directorio completo.

Opciones :

-a	<i>Es la combinación de -d (mantiene los enlaces) y -R (es recursivo).</i>
-i	<i>Interactivo pregunta por cada archivo que se copia.</i>
-p	<i>Mantiene los permisos, dueño y grupo de los archivos/directorios.</i>
-v	<i>Verbose (muestra lo que se está copiando).</i>

Ejemplos :

```
# cp mi_archivo mi_archivo_2
```

Podemos usar los comodines (***** y **?**). El ***** nos permite un conjunto de caracteres alfanuméricos o un valor alfanumérico.

```
# cp *.html web
# cp mi_?rchivo mi_archivo_3
```

Mostrar contenido de archivos: *more*, *less* y *cat*

Tanto el comando *more* y *less* nos muestra el contenido del archivo y solo podemos bajar de a una línea por vez (con la tecla *enter*) o avanzar por página (con la tecla *space*).

El comando *less* tenemos que instalarlo como paquete *apt-get install less*. A su vez, el comando *less* nos permite realizar búsquedas dentro del archivo. También con las teclas (*pgdown*) podemos bajar a la página siguiente y con la tecla (*pgup*) podemos volver a la

página anterior.

Ejemplos :

```
# more /etc/passwd
# less /etc/passwd
```

Con el comando **cat** nos permite ver el contenido pero la diferencia con **more** o **less** se nos va de pagina.

Redirección: >, >>, <

A través de la redirección podemos tomar la salida de un programa y enviarla automáticamente a un archivo. Este proceso lo maneja la propia **shell** en un lugar del programa.

La redirección se divide en tres clases: *salida a un archivo, añadir al final de un archivo, o envió de un archivo como entrada*. Para recoger la salida de un programa en un archivo, finalice la línea del comando con el símbolo mayor que (>) y el nombre del archivo en el cual quiere guardar la salida redirigida.

Ejemplos :

```
# cat /etc/passwd > salida_password_group.txt
# echo "-----" >>
salida_password_group.txt
# cat /etc/group >> salida_password_group.txt
# ls -l / > directorio.txt
# grep 'root' < /etc/passwd
```

Utilización de pipes

Los pipes son un mecanismo por el cual la salida de un programa se puede enviar como entrada de otros programas. Los programas individuales se pueden encadenar juntos para convertirse en unas herramientas extremadamente potentes.

Ejemplos :

```
# cat /etc/passwd | more

# ls -l / | more

# cat /etc/passwd | grep root | cut -d ':' -f1
```

Si quisiera concatenar comandos de forma que no tenga que esperar el resultado de un comando para ejecutar el siguiente, lo que puede hacer es escribirlos todos separándolos con el símbolo “punto y coma” (;) o con &&.

Ejemplos :

```
# cd /etc ; ls -l
```

Al utilizar (;) si da error el primer comando sigue con el proximo comando.

```
# cd /e ; ls -l
```

Al utilizar (&&) si da error el primer comando no sigue con el próximo comando.

```
# cd /etc && ls -l

# cd /e && ls -l
```

Eliminar archivo o directorios: *rm*

El comando *rm* nos permite borrar tanto archivos como directorios completo.

Opciones :

-i	<i>Antes de borrar pregunta en forma interactiva.</i>
-f	<i>En forma forzada borra sin importar si contiene o no archivos.</i>

-r o -R	<i>Forma recursiva borra tanto archivos como directorios y subdirectorios.</i>
-v	<i>Verbose (muestra lo que se está borrando).</i>

Ejemplos :

```
# rm mi_archivo_2
```

Si como *root* (és decir super usuario) ejecutamos el comando **rm -rf /** nos borrara casi todo el sistema operativo con esto romperá el sistema.

Mover archivos/directorios o renombrarlos: *mv*

Con este comando podremos mover archivos/directorios o renombrarlos.

Opciones :

-i	<i>Antes de mover pregunta en forma interactiva.</i>
-f	<i>En forma forzada mueve sin importar si contiene o no archivos.</i>
--backup	<i>Me crea un backup del archivo destino antes de ser movido.</i>
-v	<i>Verbose (muestra lo que se está borrando).</i>

Ejemplos :

```
# mv mi_archivo /home/pablo
```

```
# mv directorio_1 directorio_2
```

Creación de un directorio: *mkdir*

Con este comando nos permite crear directorio/subdirectorio.

Opciones :

-p	<i>Me crea tanto el directorio padre como los subdirectorios.</i>
-m, --mode=MODE	<i>Esto nos permite crear un directorio/subdirectorio con cierto permisos específicos.</i>
-v	<i>Verbose (muestra lo que se está borrando).</i>

Ejemplos :

```
# mkdir archivo
# mkdir /home/pablo/Documentos
# mkdir -p /Cosas/Archivos
```

En este ejemplo me crear dentro del directorio *Datos_Generales* los subdirectorios : *Archivos* , *Documentos*, *Fotos* y *Imagenes*.

```
# mkdir -p /Datos_Generales/{Archivos,Documentos,Fotos,Imagenes}
```

En este ejemplo me crea dentro del directorio *Datos* los subdirectorios : *Datos1* y *Datos2*, y dentro de estos mismos *Datos_Generales*.

```
# mkdir -p Datos/{Datos1,Datos2}/Datos_Generales
```

Creación de un directorio: *rmdir*

Una única opción es **-p**, que elimina el directorio padre también.

Ejemplos :

```
# rmdir -p Datos
# rmdir -p /Datos_Generales
```

Borrar la pantalla

Con el comando ***clear*** borramos la pantalla.

Salir de la terminal

Con el comando ***logout*** nos deslogueamos de la terminal.

Práctica

1. Entrar como usuario (**utn**).
2. Ver todos los archivos inclusive los ocultos.
3. Como se cuales son los archivos ocultos.
4. Crear un directorio padre e hijo llamado **padre/hijo**.
5. Entrar al directorio **hijo**.
6. Crear un archivo vacío dentro del directorio **hijo** llamo **pablo**.
7. Poner '**HOLA PABLO**' dentro del archivo **pablo**.
8. Volver a un directorio atrás.
9. Crear otro archivo llamo **juan**.
10. Copiar el archivo **pablo** dentro del directorio **padre**.
11. Ver el contenido del archivo **pablo** dentro del directorio **hijo**.
12. Borrar el archivo **pablo** dentro de **hijo**.
13. Borrar el directorio **hijo**.
14. Borrar el directorio **padre**.

Solución de la Práctica

1. Entrar como usuario (**utn**).
2. Ver todos los archivos inclusive los ocultos (**ls -la**).
3. Como se cuales son los archivos ocultos (**.**).
4. Crear un directorio padre e hijo llamado **padre/hijo** (**mkdir -p padre/hijo**).
5. Entrar al directorio **hijo** (**cd padre/hijo**).
6. Crear un archivo vacío dentro del directorio **hijo** llamado **pablo** (**touch pablo**).
7. Poner 'HOLA PABLO' dentro del archivo **pablo** (**echo "HOLA MUNDO" > pablo**).
7. Volver a un directorio atrás (**cd ..**).
8. Crear otro archivo llamado **juan** (**touch juan**).
9. Copiar el archivo **pablo** dentro del directorio **padre** (**cp hijo/pablo .**).
10. Ver el contenido del archivo **pablo** dentro del directorio **hijo** (**cat hijo/pablo**).
11. Borrar el archivo **pablo** dentro de **hijo** (**rm hijo/pablo**).
12. Borrar el directorio **hijo** (**rm -rf hijo**).
13. Borrar el directorio **padre** (**rm -rf padre**).