

Comandos en Linux (estructura de directorios)

Juan M. Alberola

1. Int rprete de Comandos Shell

El int rprete de comandos Shell es un programa que recibe lo que se est  escribiendo en el terminal y lo convierte en instrucciones para el sistema operativo. Cuando el usuario escribe una  rden, el int rprete ejecuta esta orden. Para ello utilizaremos el Shell de Ubuntu, que lo podemos encontrar escribiendo “terminal” en el lanzador, o simplemente navegando hasta *Aplicaciones* y despu s *Accesorios*.

Los comandos tienen la siguiente sintaxis:

```
$ nombre_comando arg1 arg2 ... argn
```

Por tanto, en la l nea de comandos escribiremos el nombre del comando seguido de uno o varios argumentos seg n necesite dicho comando. A continuaci n veremos los m s utilizados.

1.1. Comando ls

El comando `ls` lista el conjunto de ficheros, el contenido del directorio, el contenido de un  rbol de directorios o cualquier combinaci n de los anteriores. Prueba a ejecutar en tu directorio de usuario el comando `ls` y observa c mo se muestran los ficheros y directorios.

```
juanni@portatil:~$ ls
Descargas  Escritorio  Im genes  Plantillas  Videos
Documentos examples.desktop M sica    P blico
```

Figura 1: Comando “ls”

Prueba a salir de tu directorio ra z y a moverte al directorio anterior a “home” (es decir, al directorio “/”). Aqu  podemos ver diferentes carpetas del sistema (Figura 2).

```
juanni@portatil:/$ ls
bin    dev  initrd.img  media  proc  sbin  sys  var
boot  etc  lib         mnt    root  selinux  usr  vmlinuz
cdrom  home lost+found  opt    run   srv    usr
```

Figura 2: Carpetas del sistema

Si adem s queremos mostrar alguna informaci n m s detallada de los mismos, a adiremos el par metro `-l` (Figura 3).

```

juanmi@portatil:~$ ls -l
total 44
drwxr-xr-x 2 juanmi juanmi 4096 dic 28 11:00 Descargas
drwxr-xr-x 2 juanmi juanmi 4096 dic 28 11:00 Documentos
drwxr-xr-x 2 juanmi juanmi 4096 dic 28 11:00 Escritorio
-rw-r--r-- 1 juanmi juanmi 8445 dic 28 10:42 examples.desktop
drwxr-xr-x 2 juanmi juanmi 4096 dic 28 11:00 Im genes
drwxr-xr-x 2 juanmi juanmi 4096 dic 28 11:00 M sica
drwxr-xr-x 2 juanmi juanmi 4096 dic 28 11:00 Plantillas
drwxr-xr-x 2 juanmi juanmi 4096 dic 28 11:00 P blico
drwxr-xr-x 2 juanmi juanmi 4096 dic 28 11:00 Videos
juanmi@portatil:~$

```

Figura 3: Comando “ls -l”

```

juanmi@portatil:~$ ls -a
.          .dmrc          .gtk-bookmarks  P blico
..         Documentos      .gvfs          .pulse
.bash_history  Escritorio    .ICEauthority  .pulse-cookie
.bash_logout  examples.desktop  Im genes      .ssh
.bashrc       .gconf        .local         Videos
.cache        .gksu.lock    .mission-control .Xauthority
.config       .gnome2       M sica         .xsession-errors
.dbus         .goutpstream-DJ2R8W  Plantillas     .xsession-errors.old
Descargas    .goutpstream-YPP87W  .profile
juanmi@portatil:~$

```

Figura 4: Comando “ls -a”

Si queremos mostrar los elementos ocultos del sistema, pondremos al par metro `-a`. Obs rvalo dentro de tu directorio ra z (Figura 4).

Tambi n podemos combinar varios par metro en uno mismo. Prueba a hacer un `ls -l -la` o directamente `ls -la`.

Con el car cter `*` podemos hacer un listado recursivo. Prueba a hacer `ls *` sobre el directorio y observa la diferencia. Lo que hace el `*` es un listado de todos los directorios que a su vez est n en el directorio actual.

1.2. Caracteres comod n

Podemos utilizar el car cter `*` para sustituirlo por cualquier cadena de 0 o m s caracteres. Por ejemplo, con el siguiente comando pod amos listar ficheros que empiecen por `h` y tengan a continuaci n 0 o m s caracteres:

```
ls h*
```

Podemos utilizar el car cter `?` para sustituirlo por cualquier car cter individual. Por ejemplo, con el siguiente comando pod amos listar ficheros que empiecen por `h` y tengan exactamente 5 caracteres m s:

```
ls h?????
```

1.3. Comando cd

Para desplazarnos a un directorio, utilizaremos la sintaxis `cd nombre_directorio`. Por ejemplo, situado en el directorio ra z (`/home/nombre_usuario`), prueba a moverte al directorio m sica (Figura 5).

Para volver al directorio anterior, simplemente hay que hacer `cd ..` (Figura 6).

```
juanmi@portatil:~$ ls
Descargas  Escritorio  Im genes  Plantillas  Videos
Documentos examples.desktop  M sica    P blico
juanmi@portatil:~$ cd M sica/
juanmi@portatil:~/M sica$ ls
juanmi@portatil:~/M sica$
```

Figura 5: Comando “cd”

```
juanmi@portatil:~/M sica$ cd ..
juanmi@portatil:~$ ls
Descargas  Escritorio  Im genes  Plantillas  Videos
Documentos examples.desktop  M sica    P blico
juanmi@portatil:~$
```

Figura 6: Comando “cd”

Imagina que tenemos una estructura de directorios como la de la Figura 7. Si estamos situados en el directorio `/media/Memoria USB`, con el comando `cd ..` nos situar amos en `/media`, y si volvemos a hacer un `cd ..` nos situar amos en `/`. Si desde aqu  hacemos un `cd home/alumno` nos situar amos en el directorio `/home/alumno`. Adem s, tambi n podemos hacer todo este movimiento de una sola vez con `cd ../../home/alumno`, e incluso acceder al directorio apuntando directamente a su ruta a partir del directorio `/`, con `cd /home/alumno`.



Figura 7: Comando “cd”

1.4. Comando `mkdir` y `rmdir`

El comando `mkdir` nos permite crear un directorio vac o, para ello ejecutaremos este comando seguido del nombre del directorio que deseemos. Si hacemos un `ls` podemos ver que hemos creado ese directorio (Figure 8).

```
juanmi@portatil:~$ mkdir directorio_nuevo
juanmi@portatil:~$ ls
Descargas  Escritorio  M sica    salida2.txt
directorio_nuevo  examples.desktop  Plantillas  salida.txt
Documentos  Im genes      P blico    Videos
juanmi@portatil:~$
```

Figura 8: Creaci n de directorios

Nos podemos desplazar a este directorio y ver que está vacío (Figura 9).

```
juanmi@portatil:~$ cd directorio_nuevo/
juanmi@portatil:~/directorio_nuevo$ ls
juanmi@portatil:~/directorio_nuevo$
```

Figura 9: Creación de directorios

El comando `rmdir` nos permite borrar un directorio vacío, para ello ejecutaremos este comando seguido del nombre del directorio que deseemos. Si hacemos un `ls` podemos ver que el directorio ya no existe una vez borrado (Figure 10).

```
juanmi@portatil:~$ rmdir directorio_nuevo/
juanmi@portatil:~$ ls
Descargas  Escritorio  Imágenes  Plantillas  salida2.txt  Videos
Documentos examples.desktop  Música    Público     salida.txt
```

Figura 10: Borrado de directorios

1.5. Comando rm

El comando `rm` nos permite borrar un fichero, para ello ejecutaremos este comando seguido del nombre del directorio que deseemos. Si hacemos un `rm` de un fichero que habíamos creado podemos ver que ya no existe (Figure 11).

```
juanmi@portatil:~$ ls
Descargas  Escritorio  Imágenes  Plantillas  salida2.txt  Videos
Documentos examples.desktop  Música    Público     salida.txt
juanmi@portatil:~$ rm salida2.txt
juanmi@portatil:~$ ls
Descargas  Escritorio  Imágenes  Plantillas  salida.txt  Videos
Documentos examples.desktop  Música    Público     Videos
```

Figura 11: Borrado de directorios

Para borrar varios ficheros de un directorio, podemos utilizar el carácter `*`.

Si intentamos borrar un directorio, puede que no nos deje porque tiene subdirectorio o ficheros dentro (Figure 12). Para solucionar esto, si queremos borrar directorios y subdirectorios y además forzar el borrado, utilizaremos los parámetros `-rf` (Figure 13).

```
juanmi@portatil:~$ ls directorio_nuevo/
salida.txt
juanmi@portatil:~$ rmdir directorio_nuevo/
rmdir: fallo al borrar «directorio_nuevo/»: El directorio no está vacío
juanmi@portatil:~$
```

Figura 12: Borrado de directorios

1.6. Comando cp y mv

El comando `cp` se utiliza para copiar un fichero. La sintaxis es el comando seguido del fichero origen y el fichero destino (Figure 14). Para ello, el fichero

```

juanmi@portatil:~$ rm -rf directorio_nuevo/
juanmi@portatil:~$ ls
Descargas  Escritorio  Im genes  Plantillas  salida.txt
Documentos examples.desktop  M sica    P blico    Videos
juanmi@portatil:~$

```

Figura 13: Borrado de directorios

origen debe existir.

```

juanmi@portatil:~$ ls
Descargas  Escritorio  Im genes  Plantillas  salida.txt
Documentos examples.desktop  M sica    P blico    Videos
juanmi@portatil:~$ cp salida.txt salida_copia.txt
juanmi@portatil:~$ ls
Descargas  Escritorio  Im genes  Plantillas  salida_copia.txt  Videos
Documentos examples.desktop  M sica    P blico    salida.txt
juanmi@portatil:~$

```

Figura 14: Comando “cp”

Si queremos copiar varios ficheros a la vez a un mismo directorio, podemos emplear la sintaxis origen origen destino (Figura 15).

```

juanmi@portatil:~$ cp salida.txt salida_copia.txt directorio_nuevo/
juanmi@portatil:~$ ls directorio_nuevo/
salida_copia.txt salida.txt
juanmi@portatil:~$

```

Figura 15: Comando “cp”

Si queremos copiar todos los ficheros de un directorio a adiremos el car cter * para indicar que queremos copiar todos los ficheros de ese directorio (Figura 16).

```

juanmi@portatil:~$ ls directorio_nuevo/
salida_copia.txt salida.txt
juanmi@portatil:~$ ls directorio2/
juanmi@portatil:~$ cp directorio_nuevo/* directorio2/
juanmi@portatil:~$ ls directorio2/
salida_copia.txt salida.txt
juanmi@portatil:~$

```

Figura 16: Comando “cp”

Si adem s queremos copiar los subdirectorios de manera recurrente, a adiremos el par metro -r (Figura 17).

```

juanmi@portatil:~$ ls directorio_nuevo/
salida_copia.txt salida.txt subdirectorio
juanmi@portatil:~$ ls directorio2/
juanmi@portatil:~$ cp -r directorio_nuevo/* directorio2/
juanmi@portatil:~$ ls directorio2/
salida_copia.txt salida.txt subdirectorio
juanmi@portatil:~$

```

Figura 17: Comando “cp”

El comando mv se emplea para mover fichero. La sintaxis es similar a la del comando cp pero el resultado es que se elimina el origen una vez copiado en destino. Observa su ejecuci n (Figura 18).

```
juanmi@portatil:~/directorio_nuevo$ ls
salida_copia.txt  salida.txt  subdirectorio
juanmi@portatil:~/directorio_nuevo$ mv salida.txt salida_renombrado.txt
juanmi@portatil:~/directorio_nuevo$ ls
salida_copia.txt  salida_renombrado.txt  subdirectorio
juanmi@portatil:~/directorio_nuevo$
```

Figura 18: Comando “cp”