

UBUNTU: ADMINISTRACION DE FICHEROS Y DIRECTORIOS



Juan Carlos Navidad García
Sistemas Operativos en Red

1. ¿Cuál es el elemento Básico en los sistemas Linux?

El elemento básico de los sistemas Linux es la Shell, el intérprete de comandos.

2. ¿Qué significa la expresión "sensitive case"?

Significa que Linux sobre la entrada de texto es sensible al uso de mayúsculas y minúsculas en las letras.

Es decir, no interpreta de la misma manera una palabra escrita con mayúsculas que una escrita con minúsculas.

3. ¿Cómo se presenta el directorio raíz de Linux?

El directorio raíz de Linux se presenta como `/`

4. ¿Cómo puedo borrar un directorio que tenga contenido?

Los directorios se borran con el comando **rm**, pero este no borra directorios vacíos si no se le especifica. Por eso podemos utilizar **rm -r**, la **r** viene de recursividad, es decir, se aplica a los subdirectorios. En el caso de que no funcione, podemos utilizar **rm -rf**, añadiéndole la **f**, conseguimos forzar el borrado.

5. Directorios especiales de Linux (explicados)

Dentro del directorio raíz /, tenemos los siguientes directorios especiales:

- **Bin, Sbin:** Contiene ficheros ejecutables esenciales del sistema.
- **Boot:** Es un directorio estático e incluye todos los ejecutables y archivos que son necesarios en el proceso de arranque del sistema
- **Dev:** Archivos de unidades.
- **Etc:** Contiene ficheros de configuración y arranque del sistema.
- **Home:** Contiene los directorios personales de los usuarios del sistema.
- **Lib:** Librerías esenciales compartidas y módulos del kernel.
- **Media:** Representa el punto de montaje de todos los volúmenes lógicos que se montan temporalmente, ya sean unidades externas USB, otras particiones de disco, etc.
- **Opt:** En cierto modo vendría a ser como una extensión del directorio /usr, pero en este caso van todos aquellos archivos de solo lectura que son parte de programas auto-contenidos y que, por lo tanto, no siguen los estándares de almacenar los diferentes archivos dentro de los diferentes subdirectorios de /usr
- **Proc:** El directorio proc contiene una jerarquía de archivos especiales que representan el estado actual del kernel
- **Root:** Es la carpeta personal del superusuario.
- **Srv:** Sirve para almacenar archivos y directorios relativos a servidores que puedas tener instalados dentro de tu sistema
- **Sys:** Al igual que /proc, contiene archivos virtuales que proveen información del kernel relativa a eventos del sistema operativo. Es en cierto modo una evolución de /proc, y a diferencia de este último, los archivos se distribuyen de forma jerárquica.
- **Tmp:** Espacio para archivos temporales.
- **Usr:** El directorio /usr actualmente sirve para almacenar todos los archivos de solo lectura y relativos a las utilidades de usuario
- **Var:** Contiene los archivos de logs del sistema.

6. Explica con un ejemplo ruta relativa y absoluta

Una ruta absoluta o completa contiene la ruta completa de un directorio, por ejemplo, por ejemplo /opt/lampp/htdocs/...

Una ruta relativa hace referencia a una ubicación que es relativa a un directorio actual. Las rutas relativas utilizan dos símbolos especiales, un punto (.) y dos puntos seguidos (..), lo que significa el directorio actual y el directorio padre. Los dos puntos seguidos se utilizan para subir en la jerarquía. Un único punto representa el directorio actual.

Por ejemplo: Si me encuentro en el directorio `home/jnav/ejer/ejer2/` y quiero ir a `snap-store`,

Tendremos que salirnos del directorio `ejer2` hacia `ejer`, después de `ejer` iremos a `jnav`, de `jnav` iremos a `snap` y por último a `snap-store`

Escribiremos: `../../snap/snap-store`

```
├── Descargas
├── Documentos
├── ejer
│   ├── ejer
│   └── ejer2
├── Escritorio
├── Imágenes
│   └── 1152254.jpg
├── Música
├── Plantillas
├── Público
├── snap
│   └── snap-store
│       ├── 547
│       ├── common
│       └── current -> 547
└── Videos

15 directories, 2 files
jnav@jnav-vb:~$
```

7. Explica con ejemplos paso a paso como asignar permisos de forma numérica

Los permisos de un archivo se ven con la siguiente arquitectura

-rwx rwx rwx

El primer grupo de tres, se dirige al usuario, el segundo al grupo y el tercero a cualquiera que tenga acceso.

Estás letras significan lo siguiente:

- **r (lectura)** = 4
- **w (escritura)** = 2
- **x (ejecución)** = 1

Por lo tanto, si queremos cambiar los permisos de un fichero o directorio, basta con el comando **chmod** seguido de los números y el nombre del fichero/directorio.

Los números se ponen de la siguiente manera:

rwx rwx rwx: por cada grupo de tres, se suma el valor que tiene cada letra, en este ejemplo todos los grupos tienen los mismos privilegios (letras) así que van a tener el mismo valor, quedaría como 777.

Si un archivo tiene permisos **rw-rw---x** sería igual a **(4+2)(4+2)(1) = 661**

8. ¿Todos los usuarios pueden cambiar los permisos de los archivos?

No, solo aquellos que tenga privilegios por usuario o por grupo. A no ser que seas root, en ese caso tienes privilegios sobre todos los archivos.

9. ¿Cuáles son los ficheros de entrada y salida de Linux?

- La entrada se marca como **<**, simplemente estás indicando al archivo al que te diriges, por ejemplo, **cat < archivo.txt** (te dirá el contenido de archivo.txt)
- La salida se marca como **>**, en este caso estás indicando la creación de un nuevo archivo sobre un comando, por ejemplo, **cat archivo.txt > archivo1.txt** (creará el archivo1.txt y en su interior estará el resultado del comando cat)

10. Explica el redireccionamiento en Linux y pon ejemplos

- Con **>** se redirecciona hacía un archivo. Lo crea si no existe, si existe lo sobrescribe.
ls -l > lista.txt (La salida del comando se envía a un archivo en vez de la terminal.)
- Con **>>** se redirecciona hacía un archivo. Lo crea si no existe, si existe concatena la salida al final de este.
ps -ef >> procesos.txt (Concatena al archivo procesos.txt la salida del comando.)

11. ¿Qué archivo utilizamos en Linux para ignorar la salida de información de un comando?

ls -x /home/usuario > /dev/null/errores.txt

12. Diferencia entre who y whoami

- **who:** Imprime la información sobre los usuarios que están actualmente conectados.
- **whoami:** Imprime el nombre de usuario efectivo de ser ejecutado whoami

- 13. En la carpeta actividades, crear el directorio hola y el fichero.txt., a continuación, muestra el contenido de dicho directorio actividades y observa los permisos por defecto del directorio y el fichero creado.**

Creemos la carpeta hola y el documento fichero.txt en su interior:

```
jnav@jnav-server:~/actividades$ mkdir hola & touch fichero.txt hola/
jnav@jnav-server:~/actividades$
```

Comprobamos los privilegios asignados:

Veremos que tiene todos los privilegios de usuario y grupo, pero otros no podrán nada más que ver lo que hay, no podrán ni leer ni escribir.

```
drwxrwxr-x 2 jnav jnav 4096 nov 25 11:47 hola
```

En cuanto al archivo de texto, tendremos solo privilegios de escritura y lectura, otros usuarios no podrán nada más que leerlo.

```
jnav@jnav-server:~/actividades/hola$ ls -l
total 0
-rw-rw-r-- 1 jnav jnav 0 nov 25 11:49 fichero.txt
```

- a. Crea un fichero llamado comotu.txt (no le pongas comotu, sino tu nombre). Crea un enlace duro denominado duro.txt y crea un enlace simbólico denominado simbólico.txt.**

```
jnav@jnav-server:~/actividades$ touch juancarlos.txt
jnav@jnav-server:~/actividades$ ln juancarlos.txt duro.txt
jnav@jnav-server:~/actividades$ ln -s juancarlos.txt simbolico.txt
jnav@jnav-server:~/actividades$ ls -li
total 0
19220 -rw-rw-r-- 2 jnav jnav 0 nov 25 11:41 duro.txt
19220 -rw-rw-r-- 2 jnav jnav 0 nov 25 11:41 juancarlos.txt
19221 lrwxrwxrwx 1 jnav jnav 14 nov 25 11:43 simbolico.txt -> juancarlos.txt
jnav@jnav-server:~/actividades$
```

- b. Muestra los inodos de tu nombre.txt, duro.txt y simbólico.txt ¿Cuáles son idénticos? El enlace duro es idéntico.**

```
19220 -rw-rw-r-- 2 jnav jnav 0 nov 25 11:41 duro.txt
19220 -rw-rw-r-- 2 jnav jnav 0 nov 25 11:41 juancarlos.txt
19221 lrwxrwxrwx 1 jnav jnav 14 nov 25 11:43 simbolico.txt -> juancarlos.txt
jnav@jnav-server:~/actividades$
```

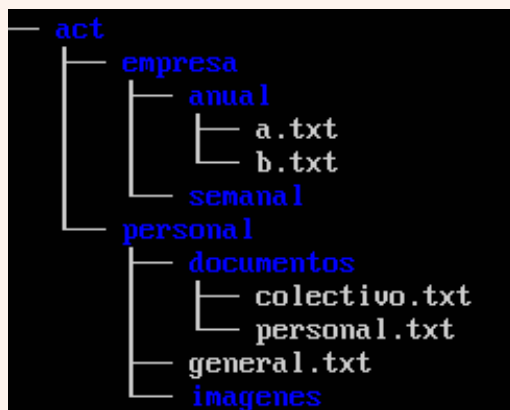
c. Borra el fichero original y observa y explica qué ha ocurrido con el enlace.

El enlace simbólico deja de funcionar y el duro sigue funcionando.

Esto se debe a que el enlace simbólico es equivalente a un enlace directo y este depende de la existencia del documento original, el enlace duro es una copia exacta del archivo original y no depende de él.

```
jnav@jnav-server:~/actividades$ rm juancarlos.txt
jnav@jnav-server:~/actividades$ ls -li
total 0
19220 -rw-rw-r-- 1 jnav jnav 0 nov 25 11:41 duro.txt
19221 lrwxrwxrwx 1 jnav jnav 14 nov 25 11:43 simbolico.txt -> juancarlos.txt
jnav@jnav-server:~/actividades$
```

14. Crea la siguiente estructura, desde tu nombre:



A continuación:

a. Muestra el directorio donde te encuentras y el árbol con la estructura construida.

```
jnav@jnav-server:~/act$ tree
.
├── empresa
│   ├── anual
│   │   ├── a.txt
│   │   └── b.txt
│   └── semanal
└── personal
    ├── documentos
    │   ├── colectivo.txt
    │   └── personal.txt
    ├── general.txt
    └── imagenes
```


- b. Estás en tu home. Accede o muestra el contenido del archivo denominado `personal.txt`. Mientras no se indique lo contrario, se utilizará ruta relativa.

```
jnav@jnav-server:~$ cat ./act/personal/documentos/personal.txt
Este es el archivo para el personal de la empresa.
jnav@jnav-server:~$ _
```

- c. Copia el archivo denominado `colectivo.txt` al directorio `anual`.

```
jnav@jnav-server:~/act$ cp ./personal/documentos/colectivo.txt ./empresa/anual
```

```
jnav@jnav-server:~/act$ ls empresa/anual/
a.txt b.txt colectivo.txt
jnav@jnav-server:~/act$
```

- d. Sitúate en el directorio `personal`. Mueve el archivo `colectivo.txt` del directorio `documentos` al directorio `empresa`, denominándose `colectivoempresarial.txt`.

```
jnav@jnav-server:~/act/personal$ mv ./documentos/colectivo.txt ../empresa/colectivoempresarial.txt
jnav@jnav-server:~/act/personal$ ls ../empresa
anual colectivoempresarial.txt semanal
jnav@jnav-server:~/act/personal$ _
```

- e. Mediante la ruta absoluta visualiza el contenido del directorio `anual`.

```
jnav@jnav-server:~$ ls /home/jnav/act/empresa/anual/
a.txt b.txt colectivo.txt
jnav@jnav-server:~$ _
```

- f. Sigues en personal, sitúate en el directorio anual.

```
jnav@jnav-server:~/act/personal$ cd ../empresa/anual
jnav@jnav-server:~/act/empresa/anual$ _
```

- g. Comando que muestre tu posición actual.

```
jnav@jnav-server:~/act/empresa/anual$ pwd
/home/jnav/act/empresa/anual
jnav@jnav-server:~/act/empresa/anual$
```

- h. Cambia los permisos al archivo a.txt para que usuario, grupo y otros solo tengan permiso de lectura.

```
jnav@jnav-server:~/act/empresa/anual$ chmod 444 a.txt
jnav@jnav-server:~/act/empresa/anual$ ls -l
total 0
-r--r--r-- 1 jnav jnav 0 nov 25 11:59 a.txt
```

- i. Desde tu posición, crear un archivo bianual.tar.gz que contenga los ficheros a.txt y b.txt

```
jnav@jnav-server:~/act/empresa/anual$ tar czvf bianual.tar.gz a.txt b.txt
a.txt
b.txt
jnav@jnav-server:~/act/empresa/anual$ ls
a.txt  bianual.tar.gz  b.txt  colectivo.txt
jnav@jnav-server:~/act/empresa/anual$ _
```

- j. Desde tu posición, mueve el archivo bianual.tar.gz al directorio empresa.

```
jnav@jnav-server:~/act/empresa/anual$ mv bianual.tar.gz ../
jnav@jnav-server:~/act/empresa/anual$ ls ../
anual  bianual.tar.gz  colectivoempresarial.txt  semanal
jnav@jnav-server:~/act/empresa/anual$
```

- k. Sigues en anual. Visualiza el contenido del archivo tar.gz.

```
jnav@jnav-server:~/act/empresa/anual$ tar xzvf ../bianual.tar.gz
a.txt
b.txt
jnav@jnav-server:~/act/empresa/anual$ _
```

- l. Desde tu posición actual, cambia la propiedad root al directorio empresa y todo su contenido.

```
jnav@jnav-server:~/act/empresa/anual$ ls -li ../../
total 8
19227 drwxrwxr-x 4 jnav jnav 4096 nov 25 13:03 empresa
19226 drwxrwxr-x 4 jnav jnav 4096 nov 25 11:58 personal
jnav@jnav-server:~/act/empresa/anual$ chmod 763 ../../empresa
jnav@jnav-server:~/act/empresa/anual$ ls -li ../../
total 8
19227 drwxrw--wx 4 jnav jnav 4096 nov 25 13:03 empresa
19226 drwxrwxr-x 4 jnav jnav 4096 nov 25 11:58 personal
jnav@jnav-server:~/act/empresa/anual$
```

- m. Cambia al directorio empresa. A continuación, extrae el contenido del archivo tar que se encuentra en dicho directorio.

```
jnav@jnav-server:~/act/empresa$ tar -xzvf bianual.tar.gz
a.txt
b.txt
jnav@jnav-server:~/act/empresa$ ls
anual a.txt bianual.tar.gz b.txt colectivoempresarial.txt semanal
jnav@jnav-server:~/act/empresa$
```

- n. Regresa a tu home y copia toda la estructura del directorio persona (incluido él) al directorio semanal.

```
jnav@jnav-server:~$ cp -r act/personal act/empresa/semanal/
jnav@jnav-server:~$ ls act/empresa/semanal
personal
jnav@jnav-server:~$ tree act/empresa/semanal
act/empresa/semanal
├── personal
│   ├── documentos
│   │   └── personal.txt
│   ├── general.txt
│   └── imagenes
3 directories, 2 files
jnav@jnav-server:~$
```

15. Crea el fichero de datos denominado datos.txt con el siguiente contenido:

```
#Nombre:Ciudad:Curso:Edad
Manuel:Segovia:2SMR:19
María:Sevilla:1SMR:16
Juan:Madrid:1SMR:18
Carmen:Murcia:2SMR:20
Julia:Badajoz:2SMR:19
Alex:Toledo:1SMR:18
Javier:Oviedo:2SMR:20
```

a. Obtén aquellos alumnos cuyo nombre empiece por M.

```
jnav@jnav-server:~$ egrep ^M datos.txt
Manuel:Segovia:2SMR:19
María:Sevilla:1SMR:16
jnav@jnav-server:~$
```

b. Busca aquellos datos de alumnos que pertenezcan a Sevilla o Badajoz.

```
jnav@jnav-server:~$ egrep "Sevilla" datos.txt && egrep "Badajoz" datos.txt
María:Sevilla:1SMR:16
Julia:Badajoz:2SMR:19
jnav@jnav-server:~$
```

c. Ordena a los alumnos por edad (de mayor a menor), y quita la línea de cabecera.

```
sort -nr orden | head -7
```

```
10:María:
jnav@jnav2:~$ sort -nr orden
20:Javier:Oviedo:2SMR
20:Carmen:Murcia:2SMR
19:Manuel:Segovia:2SMR
19:Julia:Badajoz:2SMR
18:Juan:Madrid:1SMR
18:Alex:Toledo:1SMR
16:María:Sevilla:1SMR
```

```
sort -r -k 1 orden | head -7
```

```
jnav@jnav2:~$ sort -r -k 1 orden
20:Javier:Oviedo:2SMR
20:Carmen:Murcia:2SMR
19:Manuel:Segovia:2SMR
19:Julia:Badajoz:2SMR
18:Juan:Madrid:1SMR
18:Alex:Toledo:1SMR
16:María:Sevilla:1SMR
```

- d. **Ordena a los alumnos por edad (de menor a mayor), quita la línea de cabecera y genera un nuevo fichero con dicho resultado llamado ordenados.txt**

```
sort -k 1 orden | head -7
```

```
jnav@jnav2:~$ sort -k 1 orden
16:María:Sevilla:1SMR
18:Alex:Toledo:1SMR
18:Juan:Madrid:1SMR
19:Julia:Badajoz:2SMR
19:Manuel:Segovia:2SMR
20:Carmen:Murcia:2SMR
20:Javier:Oviedo:2SMR
jnav@jnav2:~$
```

```
sort -n orden | head -7
```

```
jnav@jnav2:~$ sort -n orden
16:María:Sevilla:1SMR
18:Alex:Toledo:1SMR
18:Juan:Madrid:1SMR
19:Julia:Badajoz:2SMR
19:Manuel:Segovia:2SMR
20:Carmen:Murcia:2SMR
20:Javier:Oviedo:2SMR
```

- e. Lista ordenadamente solo las localidades diferentes que tienen los alumnos.

```
jnav@jnav2:~$ cut -d: -f1,2 orden | sort -u  
Alex:Toledo  
Carmen:Murcia  
Javier:Oviedo  
Juan:Madrid  
Julia:Badajoz  
Manuel:Segovia  
María:Sevilla  
jnav@jnav2:~$
```

- f. Mediante comando, calcula cuantos alumnos hay en 2SMR.

```
jnav@jnav2:~$ egrep "2SMR" orden | wc -l  
4
```

- g. Averigua cuantas líneas hay en el fichero datos.txt.

```
jnav@jnav2:~$ wc -l orden  
8 orden  
jnav@jnav2:~$
```

- h. Visualiza el contenido del fichero con sus líneas enumeradas.

```
jnav@jnav2:~$ cat -n orden  
1 Manuel:Segovia:2SMR:19  
2 María:Sevilla:1SMR:16  
3 Juan:Madrid:1SMR:18  
4 Carmen:Murcia:2SMR:20  
5 Julia:Badajoz:2SMR:19  
6 Alex:Toledo:1SMR:18  
7 Javier:Oviedo:2SMR:20  
jnav@jnav2:~$
```

- i. **Muestra aquellos alumnos cuya edad esté comprendida entre los 20 y los 29 años de edad.**

```
jnav@jnav2:~$ cat orden | grep "2[0-9]"
20:Carmen:Murcia:2SMR
20:Javier:Oviedo:2SMR
jnav@jnav2:~$
```

16. Redirecciona las siguientes salidas:

```
echo -e "1 Ciclo\n2 Bachillerato\n3 ESO" > text1.txt
```

```
echo -e "4\tSevilla\n5\tOviedo\n6\tMadrid" > text2.txt
```

- a. **Visualiza el contenido de los ficheros generados.**

```
jnav@jnav2:~$ cat text1.txt
"1 Ciclon2 Bachilleraton3 ESO"
jnav@jnav2:~$ cat text2.txt
"4tSevillan5tOviedon6tMadrid"
jnav@jnav2:~$
```

- b. **Lanza la siguiente instrucción y describe qué hace: `tr ' '\t' < text1.txt`**

```
jnav@jnav2:~$ tr ' '\t' < text1.txt
"1      Ciclon2 Bachilleraton3  ESO"
jnav@jnav2:~$
```

c. Redirige la entrada estándar mediante los siguientes pasos:

```
Cat << END > hola.txt  
Hola tu nombre  
END
```

```
jnav@jnav2:~$ cat << END >hola.txt  
> Hola Juan Carlos  
> END
```

d. Analiza qué ha ocurrido y visualiza el contenido del fichero.

```
jnav@jnav2:~$ cat hola.txt  
Hola Juan Carlos  
jnav@jnav2:~$
```

e. Agrega al fichero hola.txt la salida de un listado largo.

```
jnav@jnav2:~$ ls hola.txt > ls.txt  
jnav@jnav2:~$ cat ls.txt  
hola.txt  
jnav@jnav2:~$
```

f. Muestra el comando necesario para visualizar un archivo llamado nuevo.txt, de manera que, si existe su contenido, lo volcará en un fichero denominado correcto.txt, y, si no existe, en otro diferente llamado errores.txt.

```
jnav@jnav2:~$ cat nuevo.txt > errores.txt > correcto.txt  
jnav@jnav2:~$ cat nuevo.txt  
hola  
jnav@jnav2:~$ cat errores.txt  
jnav@jnav2:~$ cat correcto.txt  
hola  
jnav@jnav2:~$
```