

Sistemas informáticos.
UD 4 – Administración de sistemas operativos.
Práctica 4.3 Permisos en Linux.
Alumno: Kevin Zamora Amela

Escribe en cada ejercicio el comando introducido.

Por favor, si detectáis algún error, hacérmelo saber.

1ª.- Crear el directorio *dir2* y *dir3* en el directorio HOME ¿Cuáles son los actuales permisos del directorio *dir2*? **Dispone de los siguientes permisos: drwxrwxr-x** Y estos se interpretan de la siguiente manera: El propietario y el grupo disponen de privilegios de lectura, escritura y ejecución sobre el directorio en cuestión, mientras que otros usuarios no disponen del privilegio necesario para escribir nuevos datos.

2ª.- Utilizando la notación simbólica, eliminar todos los permisos de escritura (propietario, grupo, otros) del directorio *dir2*.

chmod u=rx dir2

chmod g=rx dir2

chmod o=rx dir2

3ª.- Utilizando la notación octal, eliminar el permiso de lectura del directorio *dir2*, al resto de los usuarios.

chmod 71 dir2

chmod 51 dir2

chmod 41 dir2

4ª.- ¿Cuáles son ahora los permisos asociados a *dir2*?

Actualmente presenta los siguientes privilegios: (d---r---x) aunque, personalmente, me chirria bastante que haya desaparecido la x, de los privilegios de ejecución del propietario y del grupo, manteniendo a su vez los permisos de lectura de 'otros'.

5ª.- Crear bajo *dir2*, un directorio llamado *dir21*.

No nos permite crear dicho subdirectorio sin volver a otorgarle más privilegios a *dir2*.

6ª.- Concederse a sí mismo permiso de escritura en el directorio *dir2* e intentar de nuevo el paso anterior.

Chmod u=wrx dir2

7ª.- ¿Cuáles son los valores por omisión asignados a los archivos?

ls -l dir2

8ª.- Cambiar el directorio actual al directorio *dir3*. Imprimir su trayectoria completa para verificar el cambio.

cd ../dir3

9ª.- ¿Cuáles son los permisos asignados en su momento a este directorio?

Dispone de los siguientes permisos: `drwxrwxr-x` Y estos se interpretan de la siguiente manera: El propietario y el grupo disponen de privilegios de lectura, escritura y ejecución sobre el directorio en cuestión, mientras que otros usuarios no disponen del privilegio necesario para escribir nuevos datos.

10ª.- Crear en el directorio propio:

El directorio *carpeta1* con los tres permisos para el propietario, dentro de él *fich1* con lectura y escritura para todos y *fich2* con lectura y escritura para el propietario y solo lectura para el resto.

<code>mkdir carpeta1</code>	<code>chmod o=r carpeta1/fich1</code>
<code>chmod u=rwx carpeta1</code>	<code>touch carpeta1/fich2</code>
<code>touch carpeta1/fich1</code>	<code>chmod u=rw carpeta1/fich2</code>
<code>chmod u=rw carpeta1/fich1</code>	<code>chmod g=r carpeta1/fich2</code>
<code>chmod g=r carpeta1/fich1</code>	<code>chmod o=r carpeta1/fich2</code>

El directorio *carpeta2* con todos los permisos para el propietario y lectura y ejecución para los del mismo grupo. Dentro *file1* con lectura y escritura para el propietario y los del grupo y *file2* con los mismos para el propietario y solo lectura para el grupo.

<code>mkdir carpeta2</code>	<code>chmod u=rw carpeta2/file1</code>
<code>chmod u=rwx carpeta2</code>	<code>chmod g=rw carpeta2/file1</code>
<code>chmod g=rx carpeta2</code>	<code>chmod u=rw carpeta2/file2</code>
<code>touch carpeta2/file1</code>	<code>chmod g=r carpeta2/file2</code>

11ª.- Visualizar la trayectoria completa del directorio actual. Crear dos directorios llamados *correo* y *fuentes* debajo del HOME.

```
pwd
/home/kzdesigner
mkdir correo
mkdir fuentes
```

12ª.- Posicionarse en el directorio *fuentes* y crear los directorios *dir1*, *dir2*, *dir3*.

```
cd fuentes
mkdir dir1
mkdir dir2
mkdir dir3
```

13ª.- Crear el directorio *menús* bajo *correo* sin moverse del directorio actual.

```
mkdir ../correo/menús
```

14ª.- Posicionarse en el directorio HOME. Borrar los directorios que cuelgan de *fuentes* que acaben en un número que no sea el 1.

Nos encontramos ya en 'home': `pwd = /home/kzdesigner`

`rm -r fuentes/????[^1]`

`rm -r fuentes/?????[1]`

Los comandos presentados son dos ejemplos, aunque estos muestran dependencia de la longitud que muestren los directorios y/o ficheros, en sus nombres. PD: Me gustaría averiguar cómo se podría construir el comando respectivo, para que así sí lograra borrar todos los elementos que cumplan con la condición del enunciado.

15ª.- Ver los permisos que tienen los archivos que empiecen por tt del directorio /dev.

`ls -l /dev/tt*`

16ª.- Obtener los permisos del archivo /usr/bin/passwd. Observas algo raro.

`-rwsr-xr-x 1 root root 59976 Feb 6 13:54 /usr/bin/passwd`

No observo nada demasiado relevante pero, por resaltar las diferentes partes/secciones que identifiqué, estas serían las siguientes:

- Privilegios del usuario 'root'
- Nombre del usuario al que pertenece la contraseña de superusuario
- Fecha de creación/instalación
- Ruta restringida del archivo resaltada y sobre fondo rojo