

PERMISOS y PROPIETARIOS DE ARCHIVOS

- 1) Crear el directorio /repo
- 2) Dentro de ese directorio crear 3 directorios al mismo nivel: stable, testing y unstable. Dar permisos totales al usuario y grupo dueño a esos 3 directorios. Para el resto quitar todos los permisos.
- 3) Crear un archivo llamado readme.md en cada subdirectorio creado. Asignar en todos los casos solamente permisos de lectura y escritura para el dueño, lectura para el grupo y nada para el resto.
- 4) Crear un archivo main.c en cada subdirectorio creado. Asignar en todos los casos todos los permisos para el dueño, lectura y ejecución para el grupo y nada para el resto.
- 5) Crear un directorio devs dentro /repo y modificar los permisos de manera tal que cualquiera pueda ingresar a /repo/devs pero sin permitir que "otros" pueda ver el contenido de /repo.
- 6) Modificar los permisos de manera tal que todos los archivos generados a partir de ahora pertenezcan al grupo **users**.
- 7) Cambiar de manera recursiva de manera que el grupo **users** sea el dueño del directorio /repo.
- 8) Crear 3 grupos de la siguiente manera:
 - a) groupadd developers; groupadd testers; groupadd auditors
- 9) Crear 4 usuarios de la siguiente manera:
 - a) useradd manager, juan; useradd maria; useradd ferb
- 10) Ejecutar los comandos correspondientes de manera recursiva para que queden de la siguiente manera:

```
/repo/
|-- [root      root    ] devs
|-- [manager  auditors] stable
|   |-- [manager auditors] main.c
|   `-- [manager auditors] readme
|-- [manager  testers ] testing
|   |-- [manager testers ] main.c
|   `-- [manager testers ] readme
`-- [manager  developers] unstable
    |-- [manager developers] main.c
    `-- [manager developers] readme
```

- 11) Agregar a juan como usuario dueño adicional de la carpeta stable
- 12) Agregar a maria como usuaria dueña adicional de la carpeta testing
- 13) Agregar a ferb como dueño adicional de la carpeta unstable
- 14) Crear un directorio dentro /repo llamado notas en el cual absolutamente cualquier usuario pueda crear archivos, pero sin borrar los de otro usuario.

REDIRECCIONES Y TUBERÍAS

 **Nota:** En los siguientes ejercicios te aconsejamos focalizarte antes en los conceptos de redireccionamiento y tubería y no tanto en los comandos utilizados.

- 1) Redireccionar la salida del comando **rpm -qa** a un archivo llamado paquetes-rpm.
- 2) Redireccionar la salida del comando **dpkg -l** a un paquete llamado paquetes-deb-
- 3) Repetir el ejercicio 1) pero redireccionando el error.
- 4) Repetir el ejercicio 2) pero redireccionando el error
- 5) Crear un paquete cpio con el comando find de todos los archivos con extensión **.png** del sistema.
- 6) Extraer los archivos del paquete cpio usado en el comando anterior.
- 7) Redireccionar la salida del comando find / -name "passwd", a un archivo llamado salida.txt y los errores a un archivo llamado errores.txt.
- 8) Ídem al ejercicio anterior, pero hacerlo de forma que redirija salida estándar y error al mismo archivo.
- 9) El comando shuf genera permutaciones aleatorias, por ejemplo si tenemos un archivo llamado letras con el siguiente contenido:

```
a
b
c
d
```

Al usar el comando shuf, podría darnos:

b
d
a
c

Pero también podríamos pedirle que tome un único elemento al azar de esta manera:

```
shuf -n1 letras
```

b

Sabiendo esto, ejecutar el comando `ls -R /etc` de manera que:

- No se vean errores pantalla
- Hacer que el comando shuf elija un archivo al azar

10) Corregir conceptualmente el comando `cut -d":" -f6 /etc/passwd > salida | xargs ls -ld`.

BONUS

TUBERÍAS Y REDIRECCIONES I

3)¿Qué devuelve:

- a)ls |cat
- b) ls > hola
- c) cat < /etc/passwd| less
- d)ls |sort

4) Si quisiéramos unir varios comandos para obtener el resultado final en el último comando , ¿usarían el siguiente ejemplo?

ls > hola | xargs -n1 file o usarian este ls |tee hola | xargs -n1 file . Justificar.

5)¿Qué hace esto ?

```
cut -d: -f1,7 /etc/passwd | awk -F: '{ print "El usuario "$1" usa la shell "$2 }'
```

6) Se ejecuta el comando siguiente:

```
find ./VirtualBox\ VMs/ -name "*"vdi" | xargs file
```

Pero da error "(No such file or directory)": ¿Cómo lo modificarías para que no de error?

PREGUNTAS DE REPASO: PROPIETARIOS

1)¿Qué comando utilizamos para cambiar el grupo y usuario de un archivo al mismo tiempo?

2)Cambiar el directorio bkp, para que le cambie el usuario dueño a root.

3)Cambiar el directorio test, para que el grupo le corresponda a root.

4)Cambiar el directorio bkp, para que recursivamente tenga como dueño root y grupo root.

5)¿Qué comando utilizamos para cambiar únicamente de grupo?

PREGUNTAS DE REPASO: REDIRECCIONES

1) ¿Qué tipo de redirecciones conocés?

2)¿Qué significa redireccionar?

- 3) Mencionar algunos ejemplos.
- 4) ¿Qué tipos de redirecciones existen?
- 5) ¿Cómo redireccionan la salida de un comando?
- 6) ¿Cómo redireccionan la salida de un comando pero guardando la salida, para luego utilizar el mismo archivo?
- 8) ¿Cómo redireccionan la salida de error que nos devuelve un comando?
- 9) ¿Cómo redireccionan la salida error y la salida estándar en archivos diferentes en un solo paso?
- 10) ¿Cómo redireccionan la salida error y la salida estándar en un mismo archivo?
- 11) ¿Cómo redireccionan la entrada estándar? Ejemplificar.
- 12) ¿Cuál es el metacaracter que sirve para encadenar comandos?
- 13) Demostrar cómo utilizarían la redirección con el carácter `<`.

PREGUNTAS DE REPASO: PERMISOS

- 1) ¿Qué letras identifican a los permisos? ¿Qué significan cada una de ellas?
- 2) ¿Qué comandos listan los permisos de un archivo? ¿Y de un directorio?
- 3) ¿Cuáles son los tres grupos en que se dividen los permisos?
- 4) ¿Qué dos formas existen para implementar los permisos?
- 5) ¿Qué sistema numérico utilizan los permisos? ¿Qué peso tienen dichas letras?
- 6) ¿Qué son los permisos especiales? ¿Cuáles son los permisos especiales existentes y para qué los usarían?
- 7) Buscar con el comando `find` los distintos archivos/directorios con permisos especiales y en cada paso buscar uno diferente.
- 8) ¿Qué comandos utilizo para cambiar los permisos?
- 9) ¿Qué parámetros utilizo para dar permisos de ejecución a un archivo?
- 10) ¿Qué parámetros sirven para dar permisos de lectura a un grupo únicamente?
- 11) ¿Qué parámetros sirven para dar permisos de lectura/escritura al grupo `others` y quitarle escritura al usuario dueño del archivo?
- 12) ¿Cuáles permisos se crean por defecto a los archivos y directorios? ¿Qué comando usamos para verificarlo?
- 13) ¿Cómo cambiamos la máscara por defecto?
- 14) ¿Cómo hacemos para que la máscara utilizada sea diferente?

PREGUNTAS DE REPASO: TUBERÍAS

- 1) ¿Qué es una tubería(pipe)?
- 2) ¿Para qué nos sirve unir los comandos con una tubería?

