

Universidad de Costa Rica
Ciencias de la Computación e informática

Fabián Orozco Chaves - B95690
Sistemas Operativos - Laboratorio 2

1. a)

R/

- **Owner:** usuario (por defecto) que **crea el archivo/directorio**.
- **Group:** grupo de usuarios con los **mismos permisos** de acceso Linux a un archivo/directorio, permite dar el mismo permiso a **varios usuarios a la vez**.
- **Other:** otro usuario que **no es el dueño ni pertenece a un grupo** (todos los demás). Tienen los permisos “de conjunto para el mundo”

1. b)

R/ El comando `chmod`, significa modo de cambio. Con este se puede **establecer permisos** (read, write, execute) en un archivo/directorio para el owner, group y other.

Sintaxis:

```
chmod permissions filename
```

2 formas de usar el comando:

1. Modo **absoluto** (numérico): los permisos de archivo no se representan como caracteres sino como un número octal de 3 dígitos.

Número	Binario	Tipo de permiso	Símbolo	Ejemplo de uso
0	000	No permission	---	1.-r-xrw-we ejecuta: <code>chmod 037 nameFile</code> 2. <code>--wxrwx</code>
1	001	Execute	-x	
2	010	Write	-w-	
3	011	Execute + Write	-wx	
4	100	Read	r--	
5	101	Read + Write	r-x	
6	110	Read + Write	rw-	
7	111	Read + Write + Execute	rwX	

2. Modo **simbólico**: puede modificar los permisos de un propietario en específico usando símbolos matemáticos.

Operador	Descripción	Denotaciones		Ejemplo de uso
+	Añade un permiso a un archivo/directorio	u	user	1. <code>-r-xrw-we</code> ejecuta: <code>chmod o=rwx nameFile</code> 2. <code>-r-xrwwe</code>
-	Quita el permiso	g	group	
=	Establece el permiso y reemplaza los permisos establecidos anteriormente	o	other	
		un	all	

1. c)

R/

- **chown**: (changing ownership) cambia el propietario del archivo/directorio (tanto usuario como grupo).
 - Sintaxis:
 - `chown user filename` ← cambia el usuario
 - `chown user:group filename` ← cambia el usuario y el grupo
- **chgrp**: (changing group) cambia solo el propietario del grupo.
 - Sintaxis:
 - `chgrp group_name filename`
- Diferencia: **chown** cambia el propietario (tanto **user** como **group** a la vez o separado) mientras que **chgrp** permite **cambiar solamente el grupo**.

1. d)

R/ La función **sudo** (superuser do or substitute user do) permite ejecutar comandos con los privilegios de seguridad de superusuario (**root**) o de otro usuario.

Privilegios de ser root:

- Cambiar los permisos, crear o eliminar cualquier/todos los archivos.
- Cambiar el nivel de ejecución del sistema.
- Modificar cuentas de usuario.
- Montar o desmontar sistemas de archivos.
- Agregar o eliminar paquetes de software.

Hace pensar más en lo que estás haciendo (con tanto poder sobre el sistema).

Sintaxis: `sudo command`

1. e)

R/ El comando **chmod 777** habilita los permisos (read, write, execute) a todas las categorías (owner, group, others) por lo que si se corre junto a un nombre de archivo, éste será de acceso público para todos.

2. a)

Agrego usuario.

```
:/home/Totis
File Edit View Search Terminal Help
[fabian:~]
$ sudo useradd Fabian_DOS
```

Agrego contraseña.

```
[fabian:~]
$ sudo passwd Fabian_DOS
Changing password for user Fabian_DOS.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

2. b)

Creo carpeta "lab2" en home.

```
[fabian:home]
$ sudo mkdir lab2

[fabian:home]
$ ll
total 0
drwx----- . 1 Fabian_DOS Fabian_DOS 80 Apr 15 20:08 Fabian_DOS
drwxr-xr-x. 1 root root 0 Apr 16 09:35 lab2
drwx----- . 1 Totis Totis 610 Apr 16 09:35 Totis
```

Cambio permisos para usuario principal.

```
[fabian:/]
$ cd home; sudo chown Totis lab2

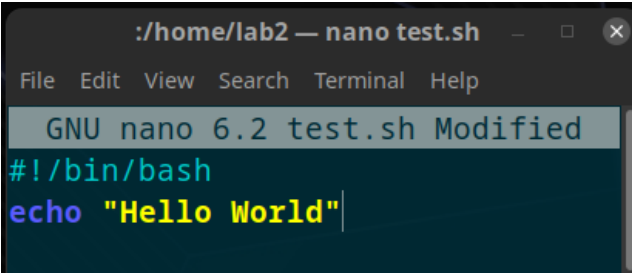
drwx----- . 1 Fabian_DOS Fabian_DOS 80 Apr 15 20:08 Fabian_DOS
drwxr-xr-x. 1 Totis root 14 Apr 16 09:38 lab2
drwx----- . 1 Totis Totis 610 Apr 16 09:35 Totis
```

2. c)

Creo documento y lo modifico con nano.

```
[fabian:lab2]
$ touch test.sh

[fabian:lab2]
$ nano test.sh
```



```
:/home/lab2 — nano test.sh
File Edit View Search Terminal Help
GNU nano 6.2 test.sh Modified
#!/bin/bash
echo "Hello World"
```

2. d)

Usuario principal, creador de “/lab2” y “test.sh”: Totis. : chmod 744

```
drwxr--r--. 1 Totis      root      14 Apr 16 09:49 lab2
-rwxr--r--. 1 Totis Totis 31 Apr 16 09:50 test.sh
```

2. e)

```
[fabian:lab2]
$ ./test.sh
Hello World
```

2. f)

El sistema no deja ejecutar el comando con el segundo usuario porque no tiene los permisos de ejecución de ese documento. Others: r--

```
[fabian:lab2]
$ sudo -u Fabian_DOS ./test.sh
sudo: unable to execute ./test.sh: Permission denied
```

2. g)

Otorgo permisos de ejecución (para “Others”) a la carpeta. chmod 745

```
[fabian:home]
$ chmod 745 lab2

[fabian:home]
$ ll
total 0
drwx-----. 1 Fabian_DOS Fabian_DOS 80 Apr 15 20:08 Fabian_DOS
drwxr--r-x. 1 Totis      root      14 Apr 16 09:49 lab2
drwx-----. 1 Totis      Totis     610 Apr 16 14:57 Totis

[fabian:lab2]
$ sudo -u Fabian_DOS ./test.sh
sudo: unable to execute ./test.sh: Permission denied
```

Al correr el comando, el sistema sigue detectando que el segundo usuario no tiene permisos de ejecución sobre el documento, por lo que el programa no se ejecuta.

2. h)

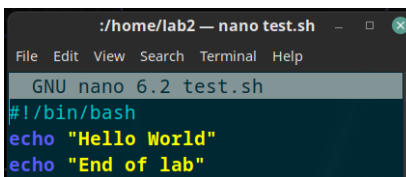
El súper usuario sí puede ejecutarlo, pues como se explicó anteriormente, tiene los privilegios de ser root.

```
[fabian:lab2]
$ sudo ./test.sh
Hello World
```

2.i)

Cambiando la línea con nano desde el usuario creador:

```
[fabian:lab2]
$ nano test.sh
```



```
:/home/lab2 — nano test.sh
File Edit View Search Terminal Help
GNU nano 6.2 test.sh
#!/bin/bash
echo "Hello World"
echo "End of lab"
```

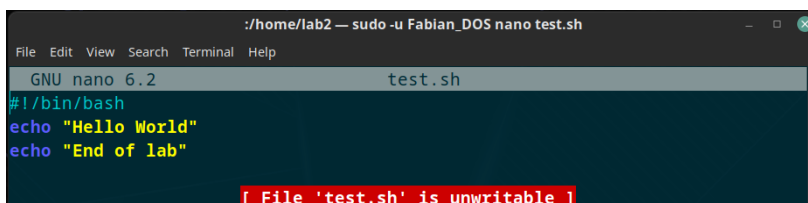
2.j)

Sí se pudo agregar la línea, debido a que el usuario creador cuenta con los permisos para hacerlo.

```
[fabian:lab2]
$ ./test.sh
Hello World
End of lab
```

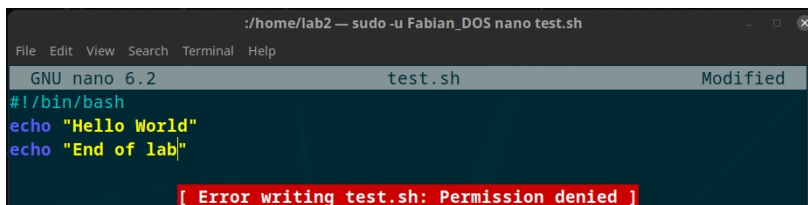
Sin embargo, para el segundo usuario, no es permitido modificar el archivo.

```
[fabian:lab2]
$ sudo -u Fabian_DOS nano test.sh
```



```
:/home/lab2 — sudo -u Fabian_DOS nano test.sh
File Edit View Search Terminal Help
GNU nano 6.2 test.sh
#!/bin/bash
echo "Hello World"
echo "End of lab"
[ File 'test.sh' is unwritable ]
```

indica que no se puede escribir.



```
:/home/lab2 — sudo -u Fabian_DOS nano test.sh
File Edit View Search Terminal Help
GNU nano 6.2 test.sh Modified
#!/bin/bash
echo "Hello World"
echo "End of lab"
[ Error writing test.sh: Permission denied ]
```

indica el error de guardado.

2.k)

Para que el segundo usuario (no creador) pueda escribir y ejecutar el archivo, deben otorgarse los permisos necesarios:

```
[fabian:lab2]
$ ll
total 4
-rwxr--r--. 1 Totis Totis 49 Apr 16 15:17 test.sh

[fabian:lab2]
$ chmod 747 test.sh
```

```
[fabian:lab2]
$ ll
total 4
-rwxr--rwx. 1 Totis Totis 49 Apr 16 15:28 test.sh
```

Ahora le es permitido modificar y ejecutar.

```
:/home/lab2 — sudo -u Fabian_DOS nano test.sh
File Edit View Search Terminal Help
GNU nano 6.2 test.sh
#!/bin/bash
echo "Hello World"
echo "End of lab"
```

```
[fabian:lab2]
$ sudo -u Fabian_DOS ./test.sh
Hello World
End of lab
```

Referencias

<https://www.guru99.com/file-permissions.html>

<https://kb.iu.edu/d/amyi#:~:text=The%20sudo%20command%20allows%20you,which%20the%20system%20administrator%20configures.>

<https://www.redhat.com/sysadmin/difference-between-sudo-su>