

Universidad de Costa Rica  
Facultad de Ingeniería  
Escuela de Ingeniería Eléctrica  
IE0117 – Programación Bajo Plataformas Abiertas  
II ciclo 2023

## Laboratorio 1 - Introducción, archivos, directorios, usuarios, grupos, y permisos

Oscar Porras Silesky C16042  
Grupo 001

Profesor: Ing. Julián Gairaud

Asistente: Felipe Agüero Peralta

4 de septiembre de 2023

## 1. Primera Parte.

1. (2 pts) Investigue qué son archivos ocultos en Unix/Linux. Explique la diferencia entre el comando “ls” y “ls -a” ¿qué resultado obtuvo al ejecutarlos?

Los archivos ocultos en sistemas Unix y Linux son archivos que se esconden para minimizar el desorden visual en los directorios y prevenir alteraciones o eliminaciones accidentales, pero siguen siendo totalmente accesibles si se conoce cómo localizarlos o listarlos con los comandos adecuados.

El comando ls en sistemas Unix y Linux se usa para listar los archivos y directorios en un directorio específico. Por lo tanto, este comando no muestra los archivos y directorios ocultos.

El comando ls -a, por el contrario, también lista todos los archivos y directorios, incluidos los ocultos (a excepción de . que representa el directorio actual y .. que representa el directorio padre).

Para comprobar el uso de los comandos ls y ls -a, se crearán 2 archivos con el comando “touch”, uno llamado “nooculto.txt” y otro llamado “.oculto.txt”

```
oscar@oscar-vbox:~$ touch Documents/nooculto.txt
oscar@oscar-vbox:~$ touch Documents/.oculto.txt
oscar@oscar-vbox:~$ ls
Desktop  Downloads  Pictures  shared    Templates
Documents Music      Public    snap      Videos
oscar@oscar-vbox:~$ cd Documents
oscar@oscar-vbox:~/Documents$ ls
nooculto.txt
oscar@oscar-vbox:~/Documents$ ls -a
.  ..  nooculto.txt  .oculto.txt
```

Figura 1: Archivos oculto y no oculto con comando ls y ls -a.

2. (3 pts) Cree un enlace simbólico ubicado en la dirección “/home/¡su usuario!;/Desktop/” que redirija hacia el archivo “/home/¡su usuario!;/.myconfig”. Demuestre que el enlace funciona al escribir cualquier cosa en el archivo (DESDE EL ENLACE), y luego imprimiendo en pantalla el contenido del archivo original.

```
oscar@oscar-vbox:~$ touch /home/oscar/.myconfig
oscar@oscar-vbox:~$ ln -s /home/oscar/.myconfig /home/oscar/Desktop
oscar@oscar-vbox:~$ nano /home/oscar/Desktop/.myconfig
oscar@oscar-vbox:~$ cat /home/oscar/Desktop/.myconfig
Confirmacion de enlace simbolico
oscar@oscar-vbox:~$ cat /home/oscar/.myconfig
Confirmacion de enlace simbolico
oscar@oscar-vbox:~$ s
```

Figura 2: Enlace simbólico creado.

Como se puede ver en la imagen, primero se crea el archivo con “touch /home/oscar/.myconfig”, luego para crear el enlace simbólico se utiliza “ln -s /home/oscar/.myconfig /home/oscar/Desktop”. Posteriormente, se confirma la creación de dicho enlace con el comando nano y se visualiza con el comando cat en ambas direcciones.

3. (2 pts) Por último elimine el archivo “.myconfig” (del punto 2.) ¿qué sucede con el enlace simbólico? ¿también se elimina? ¿qué sucede si intento leer el archivo desde el enlace?

```
oscar@oscar-vbox:~$ ls -a /home/oscar/Desktop/
.  ..  .myconfig
oscar@oscar-vbox:~$ rm /home/oscar/.myconfig
oscar@oscar-vbox:~$ ls -a /home/oscar/Desktop/
.  ..  .myconfig
oscar@oscar-vbox:~$ ^C
oscar@oscar-vbox:~$ cat /home/oscar/Desktop/.myconfig
cat: /home/oscar/Desktop/.myconfig: No such file or directory
oscar@oscar-vbox:~$
```

Figura 3: Visualización de comportamiento al borrar .myconfig.

Se observa que cuando el enlace simbólico funciona correctamente, el “.myconfig” que aparece en “ls -a /home/oscar/Desktop/” aparece en cyan, pero al borrar el “.myconfig” de “/home/oscar/.myconfig” el enlace se cae y se vuelve rojo, ya que no hay archivo enlazado. Si se intenta leer desde el enlace dice que no existe dicho archivo o directorio.

4. (5 pts)

a. Explique el concepto de inodo.

Un inodo es una estructura de datos única para cada archivo o directorio que guarda datos esenciales como permisos, propietario, tamaño y tipo de archivo.

b. ¿Cuál es su función dentro del sistema de archivos?

La función del inodo es actuar como una “tabla de contenido” que almacena todos los datos relacionados con un archivo o directorio, permitiendo al sistema operativo consultar esta información para encontrar y gestionar eficientemente los archivos y directorios.

c. ¿Cómo se puede aumentar el contador de enlaces del inodo?

El contador de enlaces en el inodo se puede aumentar creando enlaces duros adicionales al archivo. Cada vez que se crea un enlace duro a un archivo existente, el contador de enlaces del inodo del archivo original se incrementa en uno. Se puede utilizar el comando ln para crear un enlace duro: “ln arch\_original arch\_enl\_duro ”

d. ¿Cómo se puede saber el número de inodo de un archivo (desmuéstrelo con el comando correspondiente)?

Para encontrar el número de inodo de un archivo en un sistema Unix o Linux, se puede utilizar el comando ls con la opción -i: “ls -i /home/oscar/Desktop/pruebainodo.txt ”

```
oscar@oscar-vbox:~$ touch /home/oscar/Desktop/pruebainodo.txt
oscar@oscar-vbox:~$ ls -i /home/oscar/Desktop/pruebainodo.txt
524098 /home/oscar/Desktop/pruebainodo.txt
```

Figura 4: Prueba de comando para el inodo.

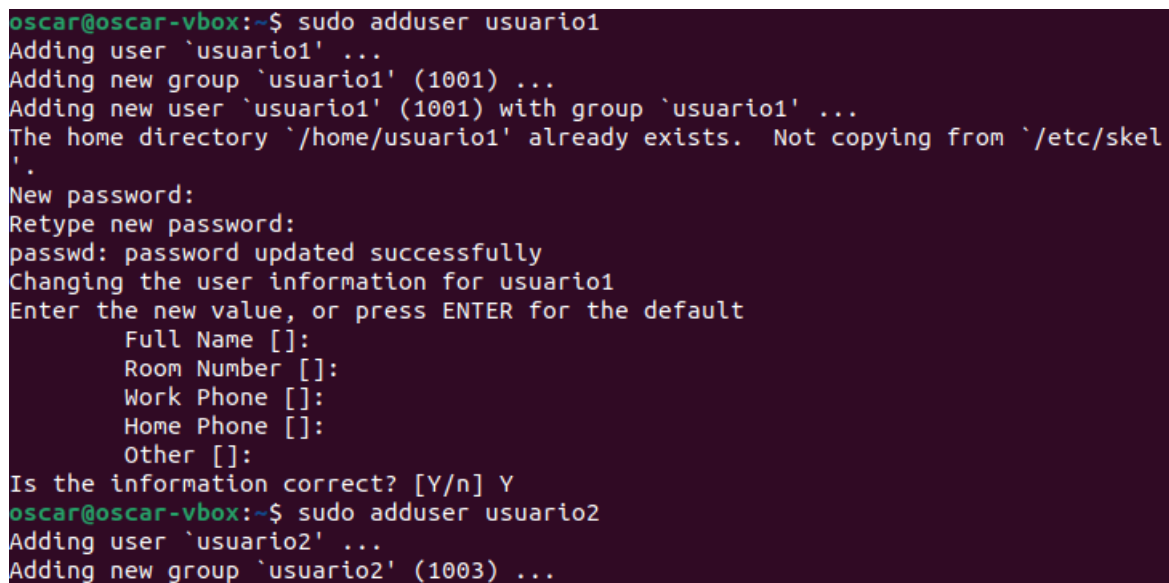
e. Explique la diferencia entre enlaces suaves (simbólicos) y enlaces duros.

Los enlaces duros aumentan el contador de enlaces en el inodo del archivo original, y por otro lado, los enlaces simbólicos tienen su propio inodo y apuntan al archivo o directorio destino por su nombre, no por su inodo. Por ejemplo, y como se vio anteriormente, si se elimina el archivo original de un enlace simbólico, el enlace se rompe y queda sin información. Si se elimina el archivo original de un enlace duro, el enlace sigue funcionando, ya que apunta al mismo espacio de memoria.

## 2. Segunda Parte.

1.1 (1 pt) Agregue 2 nuevos usuarios llamados usuario1, y usuario2 a su sistema. Utilice cualquier contraseña. Al final de este pase deberían existir al menos 3 usuarios regulares: su usuario personal, usuario1, y usuario2.

Para la creación de los usuario1 y usuario2 se utilizó el comando "sudo adduser usuario1" y "sudo adduser usuario2".

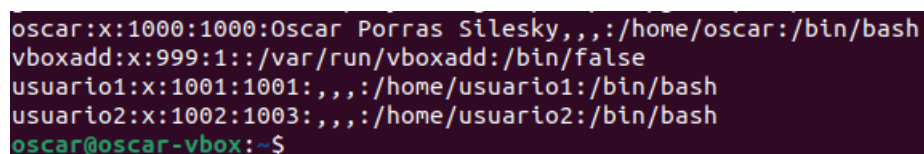


```
oscar@oscar-vbox:~$ sudo adduser usuario1
Adding user `usuario1' ...
Adding new group `usuario1' (1001) ...
Adding new user `usuario1' (1001) with group `usuario1' ...
The home directory `/home/usuario1' already exists. Not copying from `/etc/skel'.
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for usuario1
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] Y
oscar@oscar-vbox:~$ sudo adduser usuario2
Adding user `usuario2' ...
Adding new group `usuario2' (1003) ...
```

Figura 5: Creación de usuarios.

1.2 (1 pt) Describa cómo puede corroborar la existencia de dichos usuarios.

Para corroborar la existencia de dichos usuarios se utiliza el comando "cat /etc/passwd".



```
oscar:x:1000:1000:Oscar Porras Silesky,,,:/home/oscar:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
usuario1:x:1001:1001:,,,:/home/usuario1:/bin/bash
usuario2:x:1002:1003:,,,:/home/usuario2:/bin/bash
oscar@oscar-vbox:~$
```

Figura 6: Confirmación de creación de usuarios.

**2.1** (2 pt) Agregue un nuevo grupo llamado ie0117. Agregue a usuario1 a este grupo

```
oscar@oscar-vbox:~$ sudo groupadd ie0117
oscar@oscar-vbox:~$ sudo adduser usuario1 ie0117
Adding user `usuario1' to group `ie0117' ...
Adding user usuario1 to group ie0117
Done.
```

Figura 7: Confirmación de adición de usuario1 a ie0117.

**2.2** (1 pt) Describa cómo puede corroborar la existencia de dicho grupo, y también cómo comprobar que usuario1 pertenece a él. Ningún otro usuario debe pertenecer al grupo ie0117.

Se utiliza el comando "cat /etc/group".

```
usuario1:x:1001:
usuario2:x:1003:
ie0117:x:1004:usuario1
oscar@oscar-vbox:~$
```

Figura 8: Confirmación de que solo usuario1 pertenece a ie0117.

**3.** (3 pts) Cámbiese al directorio de archivos temporales del sistema y dentro de éste cree la siguiente jerarquía de archivos y directorios (recuerde que GNU/Linux diferencia entre mayúsculas y minúsculas). Utilice pwd para comprobar su localización actual. Utilice a su usuario personal para crear la jerarquía, NO utilice al usuario root para crear estos archivos y carpetas. Muestre los comandos utilizados.

```
oscar@oscar-vbox:/$ cd /tmp
oscar@oscar-vbox:/tmp$ pwd
/tmp
oscar@oscar-vbox:/tmp$ mkdir DirectorioBase
oscar@oscar-vbox:/tmp$ cd DirectorioBase/
oscar@oscar-vbox:/tmp/DirectorioBase$ mkdir Directorio1 Directorio2
oscar@oscar-vbox:/tmp/DirectorioBase$ cd Directorio1
oscar@oscar-vbox:/tmp/DirectorioBase/Directorio1$ touch Archivo1 Archivo2
oscar@oscar-vbox:/tmp/DirectorioBase/Directorio1$ cd ../
oscar@oscar-vbox:/tmp/DirectorioBase$ cd Directorio2
oscar@oscar-vbox:/tmp/DirectorioBase/Directorio2$ touch Archivo3
oscar@oscar-vbox:/tmp/DirectorioBase/Directorio2$ cd ../
oscar@oscar-vbox:/tmp/DirectorioBase$ touch Archivo4
oscar@oscar-vbox:/tmp/DirectorioBase$ cd ../
```

Figura 9: Creación de jerarquía con comandos.

```
oscar@oscar-vbox:/tmp$ find DirectorioBase
DirectorioBase
DirectorioBase/Directorio1
DirectorioBase/Directorio1/Archivo1
DirectorioBase/Directorio1/Archivo2
DirectorioBase/Directorio2
DirectorioBase/Directorio2/Archivo3
DirectorioBase/Archivo4
```

Figura 10: Confirmación de jeraquía creada.

4. (4 pts) Realice las siguientes asignaciones de usuarios y grupos, EN ORDEN:

a) Cambie el grupo del directorio DirectorioBase y todo lo que está dentro de él a ie0117 de manera recursiva (conservar el usuario). Investigue con man cómo hacerlo recursivamente.

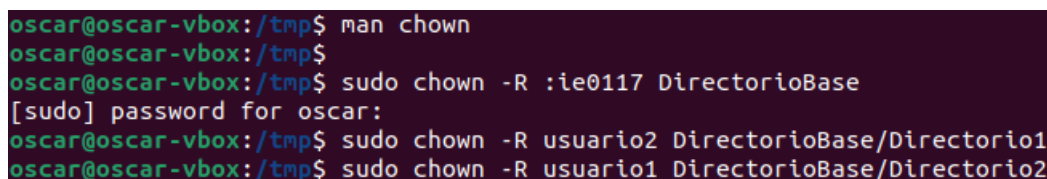
Se utiliza el comando `sudo chown -R :ie0117 DirectorioBase`.

b) Cambie el usuario dueño de Directorio1 a usuario2 (y de todos sus contenidos recursivamente).

Se utiliza el comando `sudo chown -R usuario2 DirectorioBase/Directorio1`.

c) Cambie el usuario dueño de Directorio2 a usuario1 (y de todos sus contenidos recursivamente).

Se utiliza el comando `sudo chown -R usuario1 DirectorioBase/Directorio2`.



```
oscar@oscar-vbox:/tmp$ man chown
oscar@oscar-vbox:/tmp$
oscar@oscar-vbox:/tmp$ sudo chown -R :ie0117 DirectorioBase
[sudo] password for oscar:
oscar@oscar-vbox:/tmp$ sudo chown -R usuario2 DirectorioBase/Directorio1
oscar@oscar-vbox:/tmp$ sudo chown -R usuario1 DirectorioBase/Directorio2
```

Figura 11: Asignación de grupo y usuarios.

5. (3 pts) Realice las siguientes asignaciones de permisos:

a) Archivo1: lectura y escritura al usuario, lectura y escritura al grupo, solamente lectura a los otros.

El comando a utilizar, para todos los casos, es `sudo chmod` con las direcciones correspondientes de las instrucciones.

El permiso para "lectura y escritura" se representa con el número 6 (4 para "lectura" + 2 para "escritura"). Por lo tanto, el conjunto de permisos para el "usuario", "grupo" y "otros" sería 664. Para este caso el comando es `sudo chmod 664 DirectorioBase/Directorio1/Archivo1`.

b) Archivo3: lectura, escritura y ejecución para el usuario. Solamente lectura para grupo y otros.

El permiso para "lectura, escritura y ejecución" se representa con el número 7 (4 para "lectura" + 2 para "escritura" + 1 para "ejecución"). El permiso para "solo lectura" es 4. Por lo tanto, el conjunto de permisos para el "usuario", "grupo" y "otros" sería 744. Para este caso el comando es `sudo chmod 744 DirectorioBase/Directorio2/Archivo3`.

c) Archivo4: lectura, escritura, y ejecución para el usuario, lectura y escritura para el grupo. Nada para los otros.

El permiso para "lectura, escritura y ejecución" es 7 y el permiso para "lectura y escritura" es 6. "Nada para los otros" se representa con un 0. Por lo tanto, el conjunto de permisos para el "usuario", "grupo" y "otros" sería 760.

Para este caso el comando es "sudo chmod 760 DirectorioBase/Archivo4".

```
oscar@oscar-vbox:/tmp$ sudo chmod 664 DirectorioBase/Directorio1/Archivo1
[sudo] password for oscar:
oscar@oscar-vbox:/tmp$ sudo chmod 744 DirectorioBase/Directorio2/Archivo3
oscar@oscar-vbox:/tmp$ sudo chmod 760 DirectorioBase/Archivo4
```

Figura 12: Asignación de grupo y usuarios.

6. (6 pts) Responda las siguientes preguntas, solamente tome en cuenta a usuario1, y usuario2:

A continuación, para responder la pregunta, se utilizó el comando "ls -l /direccion/" para obtener información acerca de las propiedades de los archivos y los directorios.

```
oscar@oscar-vbox:/tmp$ ls -l DirectorioBase/Directorio1/Archivo1
-rw-rw-r-- 1 usuario2 ie0117 0 sep  6 14:31 DirectorioBase/Directorio1/Archivo1
oscar@oscar-vbox:/tmp$ ls -l DirectorioBase/Directorio2/Archivo3
-rwxr--r-- 1 usuario1 ie0117 0 sep  6 14:31 DirectorioBase/Directorio2/Archivo3
oscar@oscar-vbox:/tmp$ ls -l DirectorioBase/Archivo4
-rwxrw---- 1 oscar ie0117 0 sep  6 14:31 DirectorioBase/Archivo4
oscar@oscar-vbox:/tmp$ ls -l DirectorioBase/
total 8
-rwxrw---- 1 oscar ie0117 0 sep  6 14:31 Archivo4
drwxrwxr-x 2 usuario2 ie0117 4096 sep  6 14:31 Directorio1
drwxrwxr-x 2 usuario1 ie0117 4096 sep  6 14:31 Directorio2
```

Figura 13: Propiedades de los archivos y directorios.

a) ¿Quién puede escribir al Archivo1? ¿Por qué?

Como se puede ver en la Figura 13, el dueño es el usuario 2, por lo que puede escribir. Sin embargo, el usuario1, al pertenecer al grupo ie0117, también puede escribir.

b) ¿Quién puede escribir al Archivo4? ¿Por qué?

Puede escribir el usuario1, ya que pertenece al grupo ie0117.

c) ¿Qué usuario(s) puede(n) crear archivos dentro de Directorio1? ¿Por qué?

Tanto usuario2 (que es el propietario del directorio) como usuario1 (que es miembro del grupo propietario ie0117) pueden crear archivos dentro de Directorio1. Esto se debe a que los permisos del usuario propietario y del grupo para Directorio1 son rwx, lo que otorga al propietario y a todos los miembros del grupo ie0117 derechos de lectura, escritura y ejecución en ese directorio.

7. (1 pt) Borre el directorio DirectorioBase y todos sus contenidos (tip: recursión con rm).

Se utilizó el comando "sudo rm -rf DirectorioBase" y "ls /tmp/".

```
oscar@oscar-vbox:~$ sudo rm -rf DirectorioBase
[sudo] password for oscar:
oscar@oscar-vbox:~$ ls /tmp/
snap-private-tmp
systemd-private-57babecd6dc34ff5bb08c7d1c8b6d412-color.service-TfVQGw
systemd-private-57babecd6dc34ff5bb08c7d1c8b6d412-ModemManager.service-jesadJ
systemd-private-57babecd6dc34ff5bb08c7d1c8b6d412-power-profiles-daemon.service-jRC4tY
systemd-private-57babecd6dc34ff5bb08c7d1c8b6d412-switcheroo-control.service-VUIEW9
systemd-private-57babecd6dc34ff5bb08c7d1c8b6d412-systemd-logind.service-zpF0AH
systemd-private-57babecd6dc34ff5bb08c7d1c8b6d412-systemd-oomd.service-zdv4a8
systemd-private-57babecd6dc34ff5bb08c7d1c8b6d412-systemd-resolved.service-HFbMxq
systemd-private-57babecd6dc34ff5bb08c7d1c8b6d412-systemd-timesyncd.service-Swf68A
systemd-private-57babecd6dc34ff5bb08c7d1c8b6d412-upower.service-bUWnoT
tracker-extract-3-files.1000
VMwareDnD
```

Figura 14: Eliminación de directorio y confirmación.