

Taller 10 – Sistema de Archivos en Linux y Control de Acceso

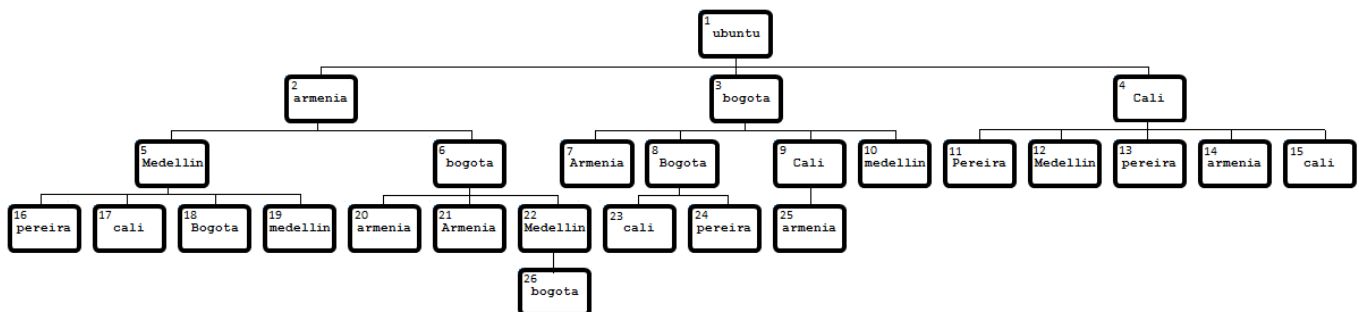
El propósito de este taller es familiarizarse con el funcionamiento del sistema de archivos y el sistema de protección en Linux.

Parte 1: Manejo de directorios

Vamos a utilizar una máquina virtual con el sistema operativo *Ubuntu Linux*. Utilice el comando **mkdir** para crear la siguiente estructura de directorios, a partir del directorio personal del usuario **ubuntu**. Tenga especial atención con las diferencias que hay entre las letras mayúsculas y las letras minúsculas. Si comete algún error, utilice el comando **rmdir** para eliminar un directorio que no corresponda con la estructura.

Es muy importante entender el uso de rutas absolutas y relativas para ubicar un archivo en la jerarquía de archivos y directorios del sistema de archivos.

Para verificar el contenido de un directorio, puede utilizar el comando **ls**. Si lo desea, puede instalar la aplicación **tree**, para que el sistema muestre la estructura de archivos y directorios.



Parte 2: Manejo de archivos

1. Utilice el comando **touch** para crear archivos, con los nombres y ubicaciones especificadas. Note que los archivos creados quedan vacíos.
 - a. En el directorio **ubuntu**:
 - **casa, carro, morro, lisa, losa, viejo, camisa y Cama.**
 - b. En el directorio **bogota** que se encuentra dentro del directorio **ubuntu**:
 - **catarata, Cama, loma, misa, Vieja, bien.**
 - c. En el directorio **Armenia** que se encuentra en el directorio **bogota** del directorio **armenia** que hace parte del directorio **ubuntu**:

- **Rata, cacerola, TASA, Mesa, bruja, buenO.**

d. En el directorio **armenia** que se encuentra dentro del directorio **Cali** del directorio **bogota** del directorio **ubuntu**:

- **tarro, perro, mama, bata, COSA, buena.**

3. Copie los siguientes archivos, usando el comando **cp**.

- Copie el archivo **loma**, al directorio **bogota**, que se encuentra en el nivel más bajo del árbol de directorios.
- Copie el archivo **TASA** al directorio padre del directorio en el que se encuentra.
- Copie el archivo **mama** al directorio padre del directorio padre del directorio en el que se encuentra.
- Copie el archivo **Cama** en el mismo directorio donde se encuentra el archivo **buenO**.
- Copie el archivo **Rata** con el nombre **roto** en el mismo directorio.

4. Cambie el nombre o mueva los siguientes archivos, utilizando el comando **mv**.

- Mueva el archivo **Mesa** al directorio **Medellin** de más bajo nivel en el árbol de directorios.
- Mueva el archivo **buena** al directorio **Cali** de más alto nivel en la jerarquía de archivos.
- Mueva el archivo **tarro** al directorio **ubuntu**, usando solamente caminos relativos, tomando como directorio actual el directorio **ubuntu**.
- Cambie el nombre del archivo **buenO** por **bueno**, usando solamente caminos relativos, tomando como directorio actual el directorio **bogota** de más bajo nivel en el árbol de directorios.
- Cambie el nombre del archivo **TASA** creado originalmente (numeral 2, ejercicio c), para que su nuevo nombre sea **TasA**.

5. Utilice el comando **rm** para eliminar los archivos a continuación.

- Borre todos los archivos del directorio **21**.
- Borre el directorio **ubuntu**.
- Borre los archivos del directorio **3** que terminan en la letra **a**.
- Tomando como directorio actual el directorio **26**, elimine el archivo **cacerola** que se

encuentra en el directorio **21**, usando un camino relativo.

- e. Borre el archivo **cama** del directorio **21**, usando un camino absoluto, teniendo como directorio actual el directorio **15**.

Parte 3: Reconocimiento del entorno

En Linux, un usuario tiene asociado un nombre y un número de identificación. Además, un usuario pertenece a un grupo primario y puede pertenecer a otros grupos. Cada grupo tiene asociado un nombre y un número de identificación.

Utilice la máquina virtual indicada por el profesor e inicie una terminal de línea de comandos.

1. Utilice el comando **id** para consultar:
 - a) Su nombre de usuario y número asignado.
 - b) El nombre de su grupo primario y número asignado.
 - c) Los nombres de los grupos a los que pertenece y sus números asignados.
 - d) Utilice el comando **echo \$HOME** para consultar su directorio de trabajo, conocido como el directorio home del usuario.

Linux ofrece la utilidad **sudo** para que un usuario regular pueda correr programas que requieren permisos de administrador (*root*). Busque información sobre este comando, entienda cuándo se usa y qué hace.

Un administrador define en el archivo de configuración de **sudo** (ubicado en la ruta **/etc/sudoers**) cuáles usuarios tienen permiso para correr **sudo** y ejecutar programas con permisos de administrador. Utilice el comando **sudo more /etc/sudoers** para revisar y analizar el contenido del archivo **sudoers**.

- e) Identifique qué parte de la configuración del archivo **/etc/sudoers** es la que le permite a su usuario ejecutar el comando **sudo** para obtener temporalmente permisos de administrador.
- f) ¿Qué ventajas ofrece el uso del comando **sudo**, comparado con crear un usuario administrador?
- g) Para el resto del taller use **sudo** si requiere correr comandos con permisos de *root*. Para correr un comando cualquiera con permisos de *root* ejecute: **sudo** seguido del comando. Por ejemplo, si desea ejecutar el comando **shutdown -h now**, debe escribir **sudo shutdown -h now**, dado que este comando está restringido al usuario *root*.

Parte 5: Sistema de Protección

El mecanismo de control de acceso en sistemas tipo Unix toma decisiones con base en los identificadores asociados al usuario o al proceso que hace un pedido de acceso y los permisos del objeto al que quiere acceder.

1. Archivos y directorios

1. El archivo **/etc/passwd** almacena información de los usuarios en un sistema Linux.
 - a) ¿Qué información se almacena acerca de los usuarios?
 - b) Ejecute el comando **ls -la /etc/passwd** y observe el resultado. ¿Quién es el propietario del archivo **passwd**?
 - c) Corra el comando **cat /etc/passwd**. Usted puede ver el contenido, aunque el archivo no es suyo. ¿Por qué?
2. El archivo **/etc/shadow** almacena información de las contraseñas de los usuarios del sistema.
 - a) Describa la diferencia entre los permisos del archivo **shadow** con el archivo **passwd**.
 - b) Cambiar la contraseña de usuario es una actividad protegida. Cada usuario puede cambiar su propia contraseña y solo *root* puede cambiar las contraseñas de otros usuarios. Cuando un usuario cambia su información (datos de usuario y/o contraseña), los archivos **passwd** y **shadow** cambian. ¿Cómo se consigue este resultado si los usuarios no tienen permisos de escritura sobre los archivos **passwd** ni **shadow**? Ayuda: observe los permisos y el *bit setuid* del programa **passwd** ubicado típicamente en la ruta **/usr/bin/passwd** e interpréte los. Si no está en esa ruta, puede usar el comando **whereis** para encontrarlo.
3. Manejo y verificación de permisos.
 - a) Permisos sobre directorios.
 - Cree un directorio llamado **ddd** en su directorio *home*. Use el comando **mkdir**.
 - Cambie los permisos para que el *dueño* del archivo tenga permisos de *lectura y escritura* (**rw**), el *grupo* tenga permisos de *lectura* (**r**) y el *resto* de usuarios tenga permisos de *lectura* (**r**). Utilice el comando **chmod**. Consulte la forma como debe usar este comando.
 - Intente entrar al directorio **ddd**.
 - ¿Cuál es el resultado? ¿Por qué?

- b) Permisos sobre directorios. Cambie los permisos del directorio **ddd** y verifique de nuevo:
- Adicione permisos de ejecución (**+x**) al dueño.
 - Intente de nuevo entrar al directorio.
 - ¿Cuál es el resultado? ¿Por qué?
- c) Permisos sobre archivos y directorios:
- Asigne los permisos **rwxr-xr-x** al directorio **ddd**.
 - Cree un archivo **f.txt** dentro del directorio. Utilice los comandos **cd ddd** y **touch f.txt**.
 - Salga del directorio y cambie los permisos del directorio **ddd** al valor **r-xr-xr-x**.
 - Liste el contenido del directorio. ¿Cuál es el resultado? ¿Por qué?
- d) Intente borrar el archivo **f.txt**. ¿Cuál es el resultado? ¿Por qué?

2. Usuarios

- a) Cree un usuario nuevo llamado **usuario2**.
- Revise el manual en línea para el comando **adduser**.
 - Cree el nuevo usuario. Necesitará permisos de *root* para correr el comando:
adduser --home /home/usuario2 -shell /bin/bash usuario2
- En este caso, **/home/usuario2** será el directorio *home* para el usuario y **/bin/bash** es el *shell* asignado al usuario.
- b) Adicione su usuario al grupo primario del usuario **usuario2**. Para esto, edite el archivo **/etc/group** y adicione su usuario al grupo correspondiente. Necesitará permisos de *root* al editar el archivo.
- c) Inicie una sesión como **usuario2**.
- Cree el archivo **g.txt**. Utilice el comando **touch g.txt** y termine la sesión.
- d) Vuelva a su usuario original e intente los siguientes comandos:
- **ls /home/usuario2/g.txt**. ¿Cuál es el resultado? ¿Por qué?

- `rm /home/usuario2/g.txt`. ¿Cuál es el resultado? ¿Por qué?

3. Consolidación de permisos

- a) Escriba un algoritmo para determinar la lista de usuarios con permisos sobre un archivo en sistemas tipo Unix (indicando el permiso específico en cada caso). El formato de cada elemento (un renglón) en la lista de respuesta es **usuario, permiso**.
- b) Escriba un algoritmo para construir la matriz de control de acceso de un sistema tipo Unix.

Para desarrollar en casa

4. Sistema de archivos

- a) Responda las siguientes preguntas sobre el sistema de archivos de Linux:
 - Busque el significado de **ext2**.
 - Busque el significado de **ext3**.
 - ¿Cuál es la principal diferencia entre **ext2** y **ext3**?
- b) Determine el propósito de los siguientes comandos (¿Para qué puede usarlos un administrador?)
 - **du**.
 - **df**.
- c) Use el comando `touch` para crear un el archivo `archivo.txt` en su directorio `home`: `touch archivo.txt`.
 - Ejecute el comando `ls -li` e identifique el número del i-nodo asociado con este archivo.
 - Ejecute el comando `stat` sobre el mismo archivo e identifique la información desplegada.
- d) Revise la representación de un directorio como archivo:
 - Cree un directorio llamado **eldirectorio**. Use el comando `mkdir eldirectorio`.
 - Utilizando un editor como **nano**, **vi** o **gedit**, abra el directorio como archivo. Por ejemplo, si utiliza el editor **vi**, puede abrir el directorio como un archivo a través del

comando **vi eldirectorio**. Para salir **:q**. Como opción, puede utilizar el comando **nano eldirectorio**, para abrir y la combinación de teclas **Ctrl+x** para salir.

5. Sistema de archivos: Enlaces simbólicos fuertes y débiles.

Linux permite crear enlaces simbólicos fuertes (*hard*) y débiles (*soft*) a un archivo. ¿Cuál es la diferencia entre estos dos tipos de enlaces simbólicos? Siga los pasos para ver la diferencia: <https://stackoverflow.com/questions/185899/what-is-the-difference-between-a-symboliclink-and-a-hard-link>.

Some nice intuition that might help, using any Linux(ish) console.

Create two files:

```
$ touch blah1; touch blah2
```

Enter some Data into them:

```
$ echo "Cat" > blah1  
$ echo "Dog" > blah2
```

(Actually, I could have used echo in the first place, as it creates the files if they don't exist... but never mind that.)

And as expected:

```
$ cat blah1; cat blah2  
Cat  
Dog
```

Let's create hard and soft links:

```
$ ln blah1 blah1-hard  
$ ln -s blah2 blah2-soft
```

Let's see what just happened:

```
$ ls -l  
  
blah1  
blah1-hard  
blah2  
blah2-soft -> blah2
```

Changing the name of blah1 does not matter:

```
$ mv blah1 blah1-new  
$ cat blah1-hard  
Cat
```

blah1-hard points to the inode, the contents, of the file - that wasn't changed.

```
$ mv blah2 blah2-new  
$ ls blah2-soft  
blah2-soft  
$ cat blah2-soft  
cat: blah2-soft: No such file or directory
```