



GNU/LINUX

Artículo III

Enamorad@ hasta los tuétanos.

Comenzamos con la diversión:

Conociendo a los gatos (cat) y gestionando archivos en Linux; permisos y propietarios.

10010.

Presentación.

Estoy muy emocionada por este tercer artículo sobre GNU/Linux, en verdad nunca imagine que llegaría hasta aquí de una forma tan rápida, tanto es mi asombro, que cuando termine con GNU/Linux me dedicare a la escritura de otros artículos :),... Luego de haber compartido con todos ustedes mis experiencias personales con GNU/Linux en los artículos anteriores, he notado cierta satisfacción personal, la cual a llegado al extremo de la adicción por liberar todos esos conocimientos que rondan por mi hermoso cerebro, ya que el conocimiento libre y espontáneo es lo único que nos libera del maldito egoísmo mundial del día a día.

En este artículo aprenderemos a cambiar los propietarios y permisos de los archivos de nuestro hermoso Sistema Operativo (S.O), utilizando como herramienta a la hermosa shell, la cual se convertirá en nuestra mejor aliada desde hoy y para siempre, es hora de comenzar con la diversión.

Conociendo a los gatos.

Antes de comenzar con la administración de archivos en GNU/Linux, debo explicar un comando sumamente importante y útil llamado “cat”. Este comando se invoca sin utilizar ningún parámetro extra y toma como entrada el teclado (entrada estándar). Veamos un simple ejemplo:

```
seneka@Annaeus:~ - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda

[seneka@Annaeus ~]$ cat
Hola como estan. Presiona ENTER
Hola como estan. Presiona ENTER
Espero que bien. Presiona ENTER
Espero que bien. Presiona ENTER
¿Como carajo salgo de este comando?(Ctrl+D)
¿Como carajo salgo de este comando?(Ctrl+D)
[seneka@Annaeus ~]$ █
```

cada vez que presionamos la tecla ENTER el comando se empeña en repetir lo que habíamos escrito. Para salir de este infierno solo presiona Ctrl+D (Control mas la tecla D), en caso de no salir a la primera presiona Ctrl+D, Ctrl+D (dos veces).

Todo lo que se escriba utilizando este comando se mostrara en nuestro monitor (salida estándar). Ahora vamos a redireccionar la salida estándar a un fichero. La sintaxis es la siguiente `cat > [nombre_del_fichero]`, sera mucho mejor si vemos un ejemplo:

```
seneka@Annaeus:~ - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda

[seneka@Annaeus ~]$ cat > fl.txt
Este es el primer ejemplo. (Presiona ENTER)
Este fichero se llama fl.txt
[seneka@Annaeus ~]$ ls
audio comprimidos Desktop documentos fl.txt
[seneka@Annaeus ~]$ █
```

para salir del comando presiona (Ctrl+D) y en caso de no lograrlo presionalo 2 veces, no te sientas ofendid@ solo me gusta recordarte las cosas :). Como puedes observar en la imagen anterior, hemos creado un fichero con el comando “cat” llamado “fl.txt” el cual contiene el siguiente texto “Este es el primer ejemplo. (Presiona ENTER)”, “Este fichero se llama fl.txt”, luego listamos (mostramos) el contenido del directorio de trabajo y observamos que se creo el fichero. ¿Pero como diablos visualizo el fichero fl.txt?, hombre (no se ofendan chicas) esto ya lo explique en el artículo anterior; “more”, “less”, pero vamos a utilizar la sintaxis del comando “cat” la cual es; `cat <nombre_fichero>`, de seguro ya lo sabias, ya te pongo un ejemplo:

```
seneka@Annaeus:~ - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda

[seneka@Annaeus ~]$ ls
audio comprimidos Desktop documentos fl.txt
[seneka@Annaeus ~]$ cat fl.txt
Este es el primer ejemplo. (Presiona ENTER)
Este fichero se llama fl.txt
[seneka@Annaeus ~]$ █
```

en realidad esto no merece la pena explicarlo, ya que es sumamente fácil de entender CIERTO.

Como puedes darte cuenta el comando “cat” nos puede servir de editor, en caso de que nuestro sistema falle y solo podamos entrar al mismo utilizando la shell. ¿pero como diablos añado mas texto al fichero fl.txt?, solo debes utilizar la siguiente sintaxis; `cat >> [nombre_fichero]`, eso si, debes de tener mucho cuidado de no confundirte; “cat >” crea un fichero y “cat >>” añade nuevo texto al fichero. Veamos un ejemplo de este ultimo:

```
seneka@Annaeus:~ - Terminal - Konsole
```

```
Sesión Editar Vista Marcadores Preferencias Ayuda
```

```
[seneka@Annaeus ~]$ ls
audio comprimidos Desktop documentos f1.txt
[seneka@Annaeus ~]$ cat >> f1.txt
Este es un nuevo texto añadido
[seneka@Annaeus ~]$ cat f1.txt
Este es el primer ejemplo. (Presiona ENTER)
Este fichero se llama f1.txt
Este es un nuevo texto añadido
[seneka@Annaeus ~]$ █
```

como puedes ver el nuevo texto se añadió al fichero “f1.txt”.
¿pero por que diablos llaman a un comando gato (cat)?, bueno en realidad el nombre de este comando viene de la abreviatura de la palabra inglesa CONCATENAR, lo cual significa UNIR. Observemos un ejemplo:

```
seneka@Annaeus:~ - Terminal - Konsole
```

```
Sesión Editar Vista Marcadores Preferencias Ayuda
```

```
[seneka@Annaeus ~]$ cat > 1.txt
texto 1
[seneka@Annaeus ~]$ cat 1.txt > 2.txt
[seneka@Annaeus ~]$ ls
1.txt audio Desktop fotos
2.txt comprimidos documentos hotel_telaraña
[seneka@Annaeus ~]$ cat 2.txt
texto 1
[seneka@Annaeus ~]$ █
```

primero creamos un fichero llamado “1.txt” que contiene el siguiente texto; **“texto 1”**, luego utilizamos el poder del comando **“cat”** para concatenar el texto a un nuevo fichero llamado “2.txt”, mostramos los archivos del directorio de trabajo y observamos que están los dos ficheros que hemos creado anteriormente, para finalizar mostramos el contenido del fichero “2.txt” y observamos que es el mismo que se encuentra en el fichero “1.txt” :).

Todavía falta explicar algo muy importante del comando **“cat”**, pero lo tendré que dejar para otro artículo, ya que la explicación del mismo requiere que los lectores conozcan como montar y desmontar un dispositivo, así que no desesperes, ya llegará el momento.

Ahora continuemos con el presente artículo, a llegado la hora de comenzar a enfrentarnos con la administración de archivos y usuarios en GNU/Linux,..... ! vamos a la ataque ;

Permisos y propietarios de los archivos.

Cada archivo y directorio de GNU/Linux posee permisos de lectura, escritura y ejecución. Aprenderemos a cambiar los permisos y los propietarios de nuestros

archivos, con el objetivo de proporcionar un mayor o menor grado de accesibilidad a los mismo. Para ver los permisos de los archivos podemos utilizar el comando **“ls -l”**, veamos un ejemplo:

```
seneka@Annaeus:~/13 - Terminal - Konsole
```

```
Sesión Editar Vista Marcadores Preferencias Ayuda
```

```
[seneka@Annaeus ~]$ ls
13 audio comprimidos Desktop documentos
[seneka@Annaeus ~]$ cd 13
[seneka@Annaeus 13]$ ls -l
total 8
drwxr-xr-x  2 seneka lp 4096 ago 29 18:47 14
[seneka@Annaeus 13]$ █
```

lo primero que hacemos es cambiar al directorio “13”, luego listamos el contenido del mismo utilizando **“ls -l”**. Podemos observar la siguiente cadena **“drwxr-xr-x”** la cual representa los permisos del directorio “14 (en color azul)”, el primer **“seneka”** significa el nombre del propietario del directorio “14”, luego observamos **“lp”** que representa el grupo al que pertenece el directorio “14”. Veamos como cambiar de propietarios.

Cambio de propietarios de archivos/directorios.

Comencemos a imaginar una situación; imaginemos que deseamos que un directorio/archivo de nuestro PC (computadora, ordenador) pase a pertenecer a otra persona (usuario):

```
seneka@Annaeus:~/13 - Terminal - Konsole
```

```
Sesión Editar Vista Marcadores Preferencias Ayuda
```

```
[seneka@Annaeus 13]$ ls -l
total 8
drwxr-xr-x  2 seneka lp 4096 ago 29 18:47 14
[seneka@Annaeus 13]$ █
```

en el directorio “13” tenemos un sub-directorio llamado “14 (en color azul)”, que pertenece al usuario **“seneka”** y al grupo **“lp”**. Queremos cambiarlo al propietario **“ada”** y al grupo **“disk”**. Para realizar los cambios tendremos que estar como root (administrador). observa el ejemplo:


```
[seneka@Annaeus ~]$ su
Password:
[root@Annaeus seneka]# ls
13 audio comprimidos Desktop documentos
[root@Annaeus seneka]# cd 13
[root@Annaeus 13]# chown ada.disk 14
[root@Annaeus 13]# ls -l
total 8
drwxr-xr-x  2 ada disk 4096 ago 29 18:47 14
[root@Annaeus 13]# █
```

como pueden ver el sub-directorio “14” ahora es del usuario “**ada**” y del grupo “**disk**”, pero sera mejor que lo explique a fondo. Primero utilizamos el comando “**su**” para volvernos “**root**”, luego listo (muestro) el contenido del usuario “**seneka**” y cambio al directorio “13”, ya que en el se encuentra el sub-directorio “14” el cual deseamos cambiar de propietario. Aquí viene lo nuevo, la utilización del comando “**chown**” una de las sintaxis es la siguiente:

chown <usuario.grupo> <archivo/directorio>

siguiendo la sintaxis. Cuando escribo “**chown ada.disk 14**”, lo que estoy es diciéndole a Linux que CAMBIE el propietario del sub-directorio “14” al usuario “**ada**” del grupo “**disk**”, así de simple. Recuerden que el punto entre usuario y grupo (ada.disk) es obligatorio.

También es posible cambiar por separado el usuario y el grupo:

“**chown ada 14**” (cambio el directorio “14” al usuario ada)

“**chown .disk 14**” (cambio el directorio “14” al grupo disk)

con esto obtenemos exactamente lo mismo, pero utilizando un paso de mas :, no olvides consultar la ayuda “**chown – help**”.

Ahora veamos otra forma de cambiar el propietario de un archivo/directorio. **¿Que no te lo crees?**, solo observa y recuerda que el objetivo de esta serie de artículos es la de aprender como funcionan las cosas y las distintas formas de hacer esas cosas :).

Siguiendo con el ejemplo anterior, vamos a volver a cambiarle el propietario al directorio “14”, en este caso el nuevo propietario volverá a ser el usuario “**seneka**” del grupo “**lp**”. Veamos el ejemplo:

```
[seneka@Annaeus ~]$ su
Password:
[root@Annaeus seneka]# ls
13 audio comprimidos Desktop documentos
[root@Annaeus seneka]# cd 13
[root@Annaeus 13]# chown seneka:lp 14
[root@Annaeus 13]# ls -l
total 8
drwxr-xr-x  2 seneka lp 4096 ago 29 18:47 14
[root@Annaeus 13]# █
```

como pueden ver el nuevo propietario es “**seneka**” del grupo “**lp**”. Para el cambio hemos utilizado lo siguiente: “**chown seneka:lp 14**”, como puedes observar la única diferencia entre el método anterior y este, es que el usuario y el grupo están separados por “: (dos puntos)”.

Veamos la sintaxis de los dos métodos:

método 1; **utilizando un punto (.) <usuario.grupo> :**

chown <usuario.grupo> <archivo/directorio>

método 2; **utilizando dos puntos (: <usuario:grupo> :**

chown <usuario:grupo> <archivo/directorio>

Si solo tienes un usuario en tu sistema, prueba con cambiar los propietarios del mismos con el administrador (root), o crea un nuevo usuario utilizando la herramienta gráfica que viene con tu GNU/Linux.

Cambiando los permisos de los archivos.

Linux es un sistema multiusuario real, por tal motivo es necesario contar con permisos de accesos a los archivos, ya que cada usuario del sistema necesita privacidad en cuanto a la información almacenada en su cuenta. Hay dos formas de cambiar los permisos de los archivos, una es utilizando letras y otra utilizando números, de mas esta decir que aprenderemos la dos formas. Es hora de comenzar con la diversión.

Dentro de Linux cada archivo posee un código de 9 bits de protección, el cual regula el acceso al mismo, clasificando a los usuarios que pueden tener acceso a los archivos en un grupo de 3 clases, el cual es el siguiente:

Clase u (user – usuario); formada por el dueño del fichero.

Clase g (group – grupo); formada por los usuarios que forman parte del grupo del propietario.

Clase o (others – otros); formada por el resto de los

usuarios.

Los 9 bits de protección de acceso a los archivos se encuentran divididos en 3 grupos de 3bits cada uno. Cada grupo de 3 bits regula el acceso a: **u**, **g**, **o** respectivamente.

Cada bit de cada grupo indica:

bit 1 (r); permiso de lectura.

bit 2 (w); permiso de escritura.

bit 3 (x); permiso de ejecución.

Para observar los permisos de cualquier archivo utilizaremos el comando **"ls -l"**. Veamos un ejemplo:

```
seneka@Annaeus:~/6 - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda

[seneka@Annaeus ~]$ ls
6 audio comprimidos Desktop documentos
[seneka@Annaeus ~]$ cd 6
[seneka@Annaeus 6]$ ls -l
total 8
-rw-r--r-- 1 seneka lp 6 ago 30 16:59 7.txt
[seneka@Annaeus 6]$
```

primero listo (muestro) el contenido del directorio de trabajo, luego cambio al directorio "6" y listo el contenido del mismo con; **"ls -l"**. Podemos observar la siguiente cadena; **"-rw-r--r--"**, la cual representa los permisos del archivo para las clases; **u**, **g**, **o** (usuarios, grupos, otros). Agrupemos esas cadenas de erres (**r**), guiones (-) y uves (**w**) en grupos de 3, comenzando por el segundo carácter de la izquierda, el cual es la primera (r). Obtenemos lo siguiente ;

rw- ; los 3 bits de permisos para el propietario del archivo "7.txt."

r-- ; los 3 bits de permisos para el grupo al que pertenece el archivo "7.txt."

r-- ; los 3 bits de permisos para el resto de los usuarios.

los guiones (-) significan; sin permiso.

teniendo esto en mente, observamos la siguiente imagen:

```
seneka@Annaeus:~/6 - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda

[seneka@Annaeus 6]$ ls -l
total 8
-rw-r--r-- 1 seneka lp 6 ago 30 16:59 7.txt
[seneka@Annaeus 6]$
```

la cual nos dice lo siguiente; el usuario **"seneka"** propietario del archivo "7.txt" tiene permisos de lectura y escritura (**rw-**). El grupo **"lp"** tiene permiso solo de lectura (**r--**) y el resto de los usuarios solo tienen permiso de lectura (**r--**). Ahora hagamos una sencilla comparación:

cadena 1:

"rw-r--r--"; cadena sacada de la unión de: (**rw-**) / (**r--**) / (**r--**), explicación anterior.

cadena 2:

"-rw-r--r--"; cadena mostrada en la imagen anterior.

Si observamos las dos cadenas nos daremos cuenta de que tienen una única diferencia, la cual es, un guión (-) al comienzo de la **"cadena 2"** (**"-rw-r--r--"**). Ese primer guión es el encargado de decirnos si estamos en un archivo o en un directorio. **¿Como me lo dice?** Observa el próximo ejemplo:

(-); estas en un archivo.

(d); estas en un directorio.

```
seneka@Annaeus:~/6 - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda

[seneka@Annaeus 6]$ mkdir 8
[seneka@Annaeus 6]$ ls -l
total 16
-rw-r--r-- 1 seneka lp 6 ago 30 16:59 7.txt
drwxr-xr-x 2 seneka lp 4096 ago 30 17:56 8
[seneka@Annaeus 6]$
```

primero creamos un sub-directorio llamado "8" dentro del directorio "6", luego listamos con: **"ls -l"**. Observamos los permisos del archivo "7.txt" y del directorio "8 (esta en color azul)". Como puedes ver, los permisos del archivo "7.txt" comienzan con un guión (-), mientras que los permisos del directorio "8" comienzan con una "d" :).

Ahora vamos a conocer el comando encargado de cambiar los permisos de los archivos, este comando es: **"chmod"**. Veamos un completo ejemplo:

```
[seneka@Annaeus ~]$ su
Password:
[root@Annaeus seneka]# ls
6 audio comprimidos Desktop documentos foto
[root@Annaeus seneka]# cd 6
[root@Annaeus 6]# ls -l
total 16
-rw-r--r-- 1 seneka lp 6 ago 30 16:59 7.txt
drwxr-xr-x 2 seneka lp 4096 ago 30 17:56 8
[root@Annaeus 6]# chmod go+w 7.txt
[root@Annaeus 6]# ls -l
total 16
-rw-rw-rw- 1 seneka lp 6 ago 30 16:59 7.txt
drwxr-xr-x 2 seneka lp 4096 ago 30 17:56 8
[root@Annaeus 6]# █
```

Bueno, primero cambiamos a modo administrador (root) y listamos (mostramos) el contenido del directorio de trabajo, luego cambiamos al directorio “6” y listamos con el comando “ls -l”, cambiamos los permisos del archivo “7.txt” con el comando “chmod go+w 7.txt” y para terminar listamos nuevamente y observamos como los permisos han sido cambiados satisfactoriamente. Ahora analicemos que hace “chmod go+w 7.txt”:

“chmod”; comando para cambiar permisos de archivos

“go+w”; g (grupo de usuarios al que pertenece el archivo), o (otros usuarios), + (añadir), w (permiso de escritura).

“7.txt”; el archivo al que se le cambiaran los permisos de acceso.

Entendiendo lo anteriormente explicado, podemos decir que cuando lanzamos el comando; “chmod go+w 7.txt”, estamos diciéndole a Linux algo como esto:

“CAMBIA los permisos del archivo “7.txt” a los GRUPOS (g) de usuarios del mismo y a los OTROS (o) usuarios AÑADIENDOLES (+) permisos de ESCRITURA (w).”

Creo que con esa explicación a quedado todo claro, ! CIERTO ¡, sin embargo tengo que explicarte algo mas.

¿Como añado o elimino un permiso de un archivo?

Con mas (+); lo añades.

Con menos (-); lo eliminas.

Bueno, si queremos volver a dejar los permisos como estaban, lo único que tendríamos que hacer seria: “chmod go-w 7.txt” y si quisiéramos eliminar la lectura (r) al grupo (g) y a los otros (o), seria algo así: “chmod go-r

7.txt”. Practica :)

La otra forma de cambiar los permisos a los archivos.

Para el siguiente método necesitaremos la siguiente tabla; en ella se nos muestra la notación en octal que debemos usar, que es una abstracción de la representación siguiente, la existencia (1) y la no existencia (0) de los tres tipos de permisos considerándolos como dígitos binarios

Binario:	Octal:
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Ya teniendo la tabla nos posicionamos en **Binario** y observamos que la misma esta formada por 8 números diferentes de 3 dígitos cada uno, representados por ceros (0) y unos (1). Imaginemos que los unos (1) activan permisos y los ceros (0) los desactivan. Como tenemos que asignar permisos al dueño del archivo; u (usuario), al grupo al que pertenece el archivo; g (grupo) y a los demás usuarios; o (otros), lo haremos aplicando a los 3 posibles permisos (lectura (r) , escritura (w), ejecución (x)) una mascara de la siguiente manera. Vemos esta tabla de ejemplo:

	Permiso Inicial	Decimal	Binario	Nuevo Permiso
Usuario:	rw-	7	111	rwX
Grupo:	r--	6	110	rw-
Otros:	r--	6	110	rw-

Ahora voy a explicar lo que hace esta simple tabla, primero nos ubicamos en **Binario** y observamos 3 números, los cuales son; **111, 110, 110**, luego nos ubicamos en **Nuevo Permiso** el cual contiene; **rwX, rw-, rw-** y observamos que los 3 datos de **Binario** y **Nuevo Permiso** están formados por 3 dígitos cada uno. Si recordamos que los unos (1) activan permisos y los ceros (0) los desactivan, entenderemos por que **111** activa los permisos de **rwX (todos)** y por que **110** solo activa los permisos de **rw**. Pero sera mejor verlo en la siguiente tabla:

Binario:	Comentario:	Permisos:
111	activa todos los permisos	rwX
110	activa los permisos (r) y (w)	rw-
110	activa los permisos (r) y (w)	rw-

Todo esto lo que quiere decir es que para activar o desactivar un permiso solo tendremos que buscar los 3 números **Binarios** correctos y luego el **Decimal** de cada uno de ellos. El decimal de **111** es **7** y el de **110** es **6**, **¿como diablos sabes eso?** pues hombre, observa la primera tabla (la mas larga :). Vamos a la practica.

```
seneka@Annaeus:/home/seneka/6 - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda

[seneka@Annaeus ~]$ su
Password:
[root@Annaeus seneka]# ls
6 audio comprimidos Desktop documentos foto
[root@Annaeus seneka]# cd 6
[root@Annaeus 6]# ls -l
total 16
-rw-r--r-- 1 seneka lp 6 ago 30 16:59 7.txt
drwxr-xr-x 2 seneka lp 4096 ago 30 17:56 8
[root@Annaeus 6]# chmod 766 7.txt
[root@Annaeus 6]# ls -l
total 16
-rwxrw-rw- 1 seneka lp 6 ago 30 16:59 7.txt
drwxr-xr-x 2 seneka lp 4096 ago 30 17:56 8
[root@Annaeus 6]#
```

Primero cambio a modo “root” con el comando “su” y listo “ls” el contenido del directorio de trabajo, luego cambio al directorio “6” y listo su contenido con “ls -l”. Observemos que los permisos del archivo “7.txt” son; “-rw-r--r--”. Luego utilizamos “chmod 766 7.txt” y cambiamos los permisos, listamos nuevamente con “ls -l” y observamos que los permisos del archivo “7.txt” cambiaron “-rwxrw-rw-”.

Veamos la sintaxis del comando “chmod”:

chmod <números> <archivo>, donde números, son los 3 números Binarios utilizando una mascara en Decimal.

Los permisos pueden ser modificados tanto en archivos como en directorios, pero debes de tener en cuenta que el concepto de *permiso de ejecución* posee un significado muy distinto para un directorio. Cambia el permiso de ejecución de un directorio y observa las diferencias :), puedes utilizar usuarios distintos y de

diferentes grupos. SUERTE.

Conclusiones.

Hemos llegado al final de este articulo, espero que ya comiences a sentir que te desenvuelves mucho mas LIBRE por tu hermoso GNU/Linux. Ya tienes conocimientos sobre la administración de permisos y propietarios, algo fundamental para el estudio de la administración de usuarios.

En el próximo articulo crearemos y eliminaremos cuentas de usuarios utilizando nuestra amada shell y continuaremos avanzando en este hermoso camino de sabiduría y aprendizaje, que solo tu espíritu rebelde y tu curiosidad podrán sortear y así llegar al final. No te detengas, no te rindas, no le tengas miedo a lo desconocido, el conocimiento es lo único que te da libertad plena, sin el seras esclavo del maldito y prostituido sistema social. Solo tu puedes elegir despertar y conocer al mundo real.

Articulo realizado por:
10010.

P.D: Gracias nuevamente a Seneka :)