

Sesión 2-3

Funciones o roles de un servidor Windows y Linux:

Servidor	MS-Windows Server 2016/2019	Linux
Servidor de Archivos	Active Directory	Samba Server, LDAP
Servidor de Base de Datos	MS SQL Server, Oracle	MySQL (MariaDB), PostgreSQL, Oracle, Firebird, MongoDB
Servidor Web	MS-IIS (Microsoft Internet Information Services), Apache	Apache, Tomcat, NGINX
Servidores DHCP (Dynamic Host Configuration Protocol)	ISC (Internet Systems Consortium)	DHCPD
Servidores DNS	BIND DNS SERVER	BIND DNS SERVER
Servidor Proxy	MS-ISA SERVER (Microsoft Internet Security and Acceleration Server)	SQUID, NGINX
Servidor Monitoreo		NAGIOS, ZABIX
Servidor Telefonía IP		ASTERISK
Servidor Correo	MS-EXCHANGE SERVER	SENDMAIL, POSTFIX, QMAIL, EXIM
Servidor Webmail		ZIMBRA, SQUIRRELMAIL, HORDE, ROUNDROBIN

Listado de Archivos y Directorios

Comando o Instrucción	Explicación o comentario
# cd /etc	Ir al directorio etc
# ls	Listado Simple
# ls gr*	Me muestra el listado de los que empiezan con "gr", búsqueda filtrada
# ls -l	Listado Largo con detalles
# ls -m	Listado (archivos, directorios, subdirectorios) separado por comas
# ls -m more	Listado (archivos, directorios, subdirectorios) separado por comas y PAGINADO (el " more" es el PAGINADOR). El paginador nos sirve para que la salida de un comando la veamos de forma "más lenta" en vez de que salga todo rápido.

Nota: El símbolo pipeline o tubería (|) lo puede insertar pulsando "Alt + 124" o pulsando el botón que esta encima de la tecla Tab de su teclado.

Comando o Instrucción	Explicación o comentario
# cd /etc	Ir al directorio etc
# ls -i	Listado de los inodos (id archivo, id directorio)
# ls -C	Listado en columnas
# ls -C more	Listado en columnas y PAGINADO (el " " more" es el PAGINADOR)
# ls -p	Listado donde a los directorios se les agregará el indicador /
# cd /	Vamos a la raíz
# ls -lR more	Listado de Directorios, Subdirectorios, con pausa, por páginas. (R viene de Recursivo, ósea nos mostrará lo que está dentro también)
# ls -x	Listado en columnas
# cd /root	Ahora nos vamos al directorio hogar del usuario root
# ls -a	Listado de los archivos y directorios ocultos y visibles. Por ejemplo: .profile .bash_history

Nota: Los archivos y directorios ocultos empiezan con un punto (.)

Ver el “historial” de comandos ingresados al sistema:

Comando o Instrucción	Explicación o comentario
# cd /root	Ahora nos vamos al directorio hogar del usuario root
# ls -a	Listado de los archivos y directorios ocultos y visibles. Por ejemplo: .profile .bash_history
# cat .bash_history	La configuración estándar de history hace que únicamente se almacenen 2000 comandos en el fichero .bash_history . Este aspecto se puede cambiar modificando la variable HISTFILESIZE
# echo \$HISTFILESIZE	Muestra el valor de la variable HISTFILESIZE
# history	El número de comandos que history puede almacenar en memoria es 1000. Por lo tanto si ejecutamos history en la terminal, aunque el fichero .bash_history contenga 2000 comandos almacenados, la terminal únicamente mostrará los últimos 1000 comandos ejecutados. Este parámetro se puede configurar mediante la variable HISTSIZE
# echo \$HISTSIZE	Muestra el valor de la variable HISTSIZE

Modificación de las variables HISTFILESIZE y HISTSIZE

Comando o Instrucción	Explicación o comentario
# cd /root	Muestra las variables de entorno del sistema.
# cat .bashrc	Vemos el contenido del archivo .bashrc Identifique: HISTSIZE=1000 HISTFILESIZE=2000
# vi .bashrc	Edite el archivo y podrá modificar los valores de HISTSIZE y HISTFILESIZE

Ver las variables de entorno del sistema:

Comando o Instrucción	Explicación o comentario
# env	Muestra las variables de entorno del sistema.

Archivos característicos dentro del directorio hogar de cada usuario (sea un usuario común o root):

Comando o Instrucción	Explicación o comentario
# cd /home	Vamos al directorio /home
#cd jeanleonardo/	Vamos al directorio hogar del usuario jeanleonardo. Para un usuario simple o común , únicamente puede hacer todo lo que quiera dentro de su directorio hogar, fuera de este sus permisos (cadena de permisos) están limitados.
# ls -la	Reconozca los archivos: .bash_history .bash_logout (En el caso del usuario root no tendrá este archivo al ir a /root) .bashrc .profile

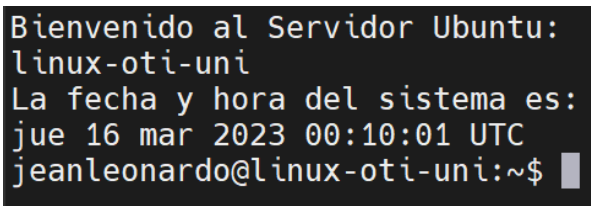
.bash_history : El archivo almacena, como su nombre ya sugiere, un historial de comandos ingresados utilizando el símbolo del sistema de Bash.

.bash_logout : es el fichero o archivo leído por Bash, cuando salimos del sistema.

.bashrc : El .bashrc contiene una serie de configuraciones para la sesión de terminal. Esto incluye la configuración o habilitación de: coloración, finalización, el historial de Shell, alias de comandos y más. Existe un archivo ~/.bashrc por cada usuario del sistema. Su configuración solo afecta al usuario que está ejecutando la sesión y dicho archivo se encuentra en el directorio \$HOME del usuario. Este archivo puede ser modificado por el usuario propietario del \$HOME donde está ubicado.

.profile : El archivo .profile se encuentra en el directorio inicial (\$HOME) y le permite personalizar el entorno de trabajo individual.

Por ejemplo, personalizemos un mensaje de bienvenida editando el archivo **.profile**:

Comando o Instrucción	Explicación o comentario
# vi .profile	Editamos el archivo .profile
Agregue en las líneas finales con mucho cuidado, lo siguiente: clear echo "Bienvenido al Servidor Ubuntu:" hostname echo "La fecha y hora del sistema es:" date	Agregado de texto al editar el archivo.
:wq!	Guardamos y salimos
Abra otra sesión, logueese con su usuario (jeanleonardo en mi caso) y que observa?	

Usuarios y Grupos:

Creación de usuarios y grupos:

Una cuenta de usuario. permite tener acceso a los recursos del sistema. Estos recursos son: (Archivos, Directorios, servicios, programas etc). **Para crear un usuario en el servidor, deberá estar con el usuario root (administrador) .**

Comando o Instrucción	Explicación o comentario
# sudo su	Para entrar como root
# adduser blopez (Berta Lopez)	Crea un usuario simple con un grupo primario de nombre igual al nombre del usuario. Nos pedirá que ingresemos la contraseña y algunos campos de información adicionales.
# passwd blopez Por ejemplo, asigne contraseña: otiuni	Sirve para cambiar el password del usuario blopez, por cualquier razón que creamos conveniente.
# cd /home	Verifiquemos que el nuevo usuario tiene su directorio hogar creado en el directorio /home
# ls -l	Se ve el directorio hogar "blopez" creado para el usuario "blopez"

Sintaxis:

```
drwxr-xr-x 2 blopez blopez 4096 mar 16 00:20 blopez
*****
Permisos Usuario Grupo Tamaño Fecha de Directorio
Dueño Propietario en bytes creacion del usuario
del del usuario
```

Loguearse con un usuario creado:

Comando o Instrucción	Explicación o comentario
Abra una nueva terminal en MobaXterm	
login as: blopez blopez@192.168.1.207's password:	
blopez@linux-oti-uni:~\$	El \$ nos indica que estamos logueados con usuario común
\$ who am i	
\$ pwd	
\$ tty	

Restricción de permisos en un usuario normal o común:

Comando o Instrucción	Explicación o comentario
\$ cd /root	Nos sale: -bash: cd: /root: Permission denied Esto debido a la restricción de permisos, porque blopez es un usuario común
\$ cd /var	Tenemos permiso de lectura al menos
\$ ls -l	Listamos el contenido
\$ mkdir compras	Permiso denegado solo puedes ver
\$ vi file	Intentemos crear un archivo
:wq!	No nos permite grabar
:q!	Solo nos queda salir

Directorio hogar de un usuario común y creación de directorios dentro de este:

Comando o Instrucción	Explicación o comentario
\$ cd	Este comando nos lleva directamente al directorio hogar del usuario (sea uno común o el root)
\$ pwd	Comprobamos
\$ cd /home/blopez	También puede usar esto en todo caso.
\$ pwd	Comprobamos
Comando o Instrucción	Explicación o comentario

\$ mkdir nomina compras mermas	Creación de 3 directorios dentro del directorio hogar del usuario "blopez"
\$ ls -l	Listado con detalles

Eliminación de un usuario común

Para eliminar a un usuario, tiene que estar seguro de hacerlo, ¿por qué?

Porque se eliminarán todos sus archivos, directorios, etc. No existe recuperación de esa cuenta del usuario. Se sugiere deshabilitar al usuario, por un tiempo. Si aun así está seguro de hacerlo, ingrese el siguiente comando estando con el usuario root.

Comando o Instrucción	Explicación o comentario
# userdel -rf blopez	Eliminación del usuario blopez y todos sus archivos, directorios, etc.
# cd /home	Comprobacion
# ls	Se eliminó el directorio hogar del usuario "blopez"

Importante: Debemos estar como root para eliminar un usuario

Creación de Grupos:

Un grupo es un conjunto de usuarios que tiene características comunes. ejem: tienen acceso a los mismos archivos, directorios, recursos, contraseñas etc. Por ejemplo, puede ser un conjunto específico de usuarios que pertenece a un área específica de una empresa.

Importante: Debemos estar como root para crear un grupo

Comando o Instrucción	Explicación o comentario
# groupadd almacen	Creación del grupo "almacen"
# cat /etc/group	Vemos el contenido del archivo /etc/group
# tail /etc/group	Vemos la 5 ultimas líneas del archivo /etc/group. Observe: almacen:x:1001: El grupo tiene como nombre "almacen" y un GID (identificador de grupo) de 1001.

Eliminación de Grupos:

Comando o Instrucción	Explicación o comentario
# groupdel nombre_de_grupo	Eliminación del grupo con nombre " nombre_de_grupo "

Importante: No confundir, una cosa es eliminar un grupo y otra es eliminar a un usuario de un grupo, esto lo veremos más adelante.

Creación de un usuario y agregado de este a un grupo ya existente o creado anteriormente:

En este caso crearemos el usuario **"jneyra"** (José Neyra) y de paso lo agregaremos al grupo **"almacen"** (grupo que creamos en pasos anteriores) y añadiremos una "Descripción" de este usuario.

Nota: Para crear un usuario en el servidor, deberá estar con el usuario root (administrador)

Comando o Instrucción	Explicación o comentario
# adduser jneyra --ingroup almacen --gecos "Usuario de almacen" Por ejemplo, asigne contraseña: otiuni	Creación del usuario jneyra, el cual será asignado al grupo almacen (este será su grupo primario) y con una descripción "Usuario de almacen"
# cd /home	Verificamos
# ls	Listamos

Logueandonos con el usuario "jneyra":

Comando o Instrucción	Explicación o comentario
Abra una nueva terminal en MobaXterm	
login as: jneyra blopez@192.168.1.207's password:	
jneyra@linux-oti-uni:~\$	El \$ nos indica que estamos logueados con usuario común
\$ who am i	
\$ pwd	
\$ tty	
\$ ls -la	Vemos la presencia de los archivos: .bash_logout .bashrc .profile
\$ ls	Listamos
\$ cd	Vamos al directorio hogar
\$ mkdir existencias importaciones mermas	Creación de 3 directorios dentro del directorio hogar del usuario "jneyra"
\$ ls -l	Comprobamos

El archivo /etc/passwd:

Este es un archivo es una base de datos de usuarios. Se encuentran los usuarios creados por el sistema operativo al momento de instalar Linux, y luego al final (o en las líneas finales) están los usuarios creados por el root.

Importante: Con un usuario común podemos ver el contenido del archivo /etc/passwd, pero si queremos editar el contenido del archivo /etc/passwd tenemos que estar con el usuario root o raíz.

Comando o Instrucción	Explicación o comentario
Cambia al usuario root con "sudo su"	
# cat /etc/passwd	
# head /etc/passwd	Muestra las primeras 5 líneas del archivo. Identifique el usuario root. Note que el root tiene asignado como ID de usuario y grupo al numero 0.
# tail /etc/passwd	Muestra las ultimas 5 líneas del archivo. Identifique los usuarios creados por el usuario root.

Sintaxis:

```

-----
Usuario      Id      Directorio
Usuario      Usuario      Particular (Directorio hogar del usuario jneyra)
-----
jneyra:x:1000:1000:USUARIO DE ALMACEN:/home/jneyra/:/bin/bash
-----
Password     Id      Comentario (GECOS)      Shell
Encriptado   Grupo

```

slice	:	x	:	1002	:	1002	:	Usuario Slice,,, :	/home/slice :	/bin/bash
										Shell INTERPRETE DE COMANDOS UTILIZADO
									Carpeta personal	Ruta de la carpeta personal.
								Información del usuario	Nombre, ubicación, teléfono del trabajo, de la oficina.	
								ID de grupo (GID)	ID del grupo principal del usuario. La información de los grupos está en /etc/groups.	
								ID de usuario (UID)	El 0 está reservado para root y 1-99 para cuentas predefinidas. 100-999 para cuentas administrativas del sistema.	
								Contraseña	Una x indica que la contraseña se encuentra encriptada en /etc/shadow. Debe tener entre 6 y 8 caracteres como mínimo.	
								Nombre de usuario	Nombre que identifica al usuario en el sistema. Debe tener entre 1 y 32 caracteres.	

Observaciones:

- La mayoría de los usuarios del sistema terminan en **/sbin/nologin**
- Los usuarios creados con "**adduser**" salen en las líneas finales del archivo **/etc/passwd**
- Vemos que los usuarios root, jeanleonardo, jneyra (estos dos ultimos creados por adduser) tienen como **shell** de usuario a **/bin/bash**

Deshabilitación del “login” de un usuario:

Si un usuario sale de vacaciones o se va de la empresa se deshabilitará por un tiempo prolongado.

Si desea deshabilitar a un usuario, deberá cambiar su Shell (campo shell del usuario en el archivo `/etc/passwd`) de `/bin/bash` por `/usr/sbin/nologin`

Importante: Como nuevamente vamos a editar el archivo `/etc/passwd`, para esto debemos estar con el usuario root o raíz.

Comando o Instrucción	Explicación o comentario
# vi /etc/passwd	Editamos el archivo
Le debe quedar algo parecido a: jneyra:x:1000:1000:USUARIO DE ALMACEN:/home/jneyra:/usr/sbin/nologin	
:wq!	Guardar y salir

Comprobamos la deshabilitación “logueandonos” con el usuario “jneyra”:

Comando o Instrucción	Explicación o comentario
Abra una nueva terminal en MobaXterm	
login as: jneyra jneyra@192.168.1.207's password:	¿Que observa? ¿Pudo loguearse como en pasos anteriores?

El archivo /etc/group:

Es un archivo donde se almacenan los grupos, grupos por default (grupos creados al instalar Linux) y lo grupos creados por el root al final.

Importante: Con un usuario común podemos ver el contenido del archivo `/etc/group`, pero si queremos editar el contenido del archivo `/etc/group` tenemos que estar con el usuario root o raíz.

Comando o Instrucción	Explicación o comentario
Cambia al usuario root con “sudo su”	
# cat /etc/group	
# head /etc/group	Muestra las primeras 5 líneas del archivo. Identifique el usuario root. Note que el grupo root tiene asignado como ID de grupo al número 0.
# tail /etc/group	Muestra las ultimas 5 líneas del archivo. Identifique los grupos creados por el usuario root.

Sintaxis:

```

      Password
      --
almacen:x:1002
-----
      Grupo      ID Grupo

```

El archivo /etc/shadow:

Es un archivo donde se guardan las contraseñas encriptadas de los usuarios. Este archivo no se debe editar, solo se puede mostrar por seguridad.

Importante: Con un usuario común podemos ver el contenido del archivo /etc/shadow, pero si queremos editar (en el caso de este archivo no se recomienda que edite el archivo pero de poder se puede) el contenido del archivo /etc/shadow tenemos que estar con el usuario root o raíz

Comando o Instrucción	Explicación o comentario
Cambia al usuario root con "sudo su"	
# cat /etc/shadow	

Sintaxis:

slice:\$1\$NLJJ6\$ow5g1l1NgYITqqQQy5D21:14234:0:99999:7: : :	
	Caducidad Días a los que se deshabilita la cuenta contados desde el 1 de enero de 1970.
	Inactivo Días a los que se deshabilita la cuenta después de que caduque la contraseña.
	Aviso Días a los que el usuario será avisado de que debe cambiar la contraseña antes de que ésta caduque.
	Máximo Días durante los que la contraseña es válida. Al terminar el usuario tiene que cambiar la contraseña.
	Mínimo Días que deben pasar como mínimo para que el usuario pueda cambiar la contraseña.
	Último cambio Días que han pasado desde la última vez que la contraseña fue cambiada contados desde el 1 de enero de 1970.
Contraseña	Contraseña encriptada. La forman entre 13 y 24 caracteres (a-z, A-Z, 0-9, \, /). Si comienza por el carácter \$, indica que la contraseña se ha encriptado usando un algoritmo distinto de DES. Si comienza por \$1\$, el algoritmo de cifrado está basado en MD5.
Nombre de usuario	Nombre que identifica al usuario en el sistema. Debe tener entre 1 y 32 caracteres.

Precisiones teóricas acerca de grupos primarios (o principal) y grupos secundarios (o adicionales):

1. El grupo primario de un usuario es el grupo por defecto con el que se asocia la cuenta, generalmente comparte el mismo nombre que el usuario o también puede ser el grupo que se le añade al usuario al crear y añadirlo a un grupo en un mismo paso, por ejm con el comando **"adduser jneyra --ingroup almacen --gecos "Usuario de almacen"** creamos el usuario jneyra y lo añadimos al grupo "almacen", en este caso para jneyra su grupo primario sería "almacen", cuando solo creamos el usuario con el comando **"adduser jneyra"** automáticamente se creará un grupo con el mismo nombre, es decir "jneyra" y en este caso para el usuario jneyra su grupo primario sería jneyra, teniendo el mismo nombre, esto se hace por defecto al usar ese comando.
2. Los directorios y los archivos que el usuario crea tendrán asignados como GRUPO PROPIETARIO al GRUPO PRIMARIO del usuario.
3. Un grupo secundario es cualquier grupo adicional del que un usuario forma parte.
4. Grupo principal: es el grupo al cual el sistema operativo asigna a los archivos creados por el usuario. Cada usuario debe pertenecer a un grupo principal.
5. Grupos secundarios: especifica uno o más grupos a los que también pertenece un usuario.
Los usuarios pueden pertenecer a hasta 15 grupos secundarios.
6. Cada usuario debe ser miembro de un grupo primario
7. Solo puede haber un grupo primario para cada usuario, es decir **un usuario solo puede tener un grupo primario, no puede tener 2 o más grupos primarios**
8. Los grupos primarios se "encuentran" en el archivo /etc/passwd, lo colocho entre comillas porque en realidad lo que se encuentra es el GID (group identifier) de los grupos primarios. Con estos GID puedo ir al archivo /etc/group y buscar que nombre tienen. En el archivo /etc/group están todos los grupos disponibles del sistema, es decir están los primarios y los secundarios de todos los usuarios (sean comunes o el root)
9. Una forma de ver rápidamente cual es el "grupo primario" y los "grupos secundarios" de un usuario es ingresando el comando "id **NOMBREDEUSUARIO**". A la salida de este comando nos saldrá el **gid** asociado al grupo primario (el gid caracteriza al grupo primario de un usuario) y en **groups** aparecerá el grupo primario (con su gid) seguido del grupo o de los grupos secundarios a los cuales pertenece el usuario.
10. ¿El UID y el GID deben tener el mismo número para un usuario? No necesariamente.
11. Recuerde que no se puede eliminar el grupo primario de un usuario existente. Para borrarlo tendría que borrar primero al usuario y luego recién borrar el grupo primario. O si desea **"borrar"** el grupo primario de un usuario, pero sin borrar al usuario, primero tendrá que cambiar de grupo primario al usuario en cuestión, después de esto ya puede "borrar" el grupo.

Diferencia entre “Password Expiration” (Expiración de Contraseña) y “Account Expiration” (Expiración o Caducación de Cuenta)

Una cosa es cuando la cuenta CADUCA (generalmente se dice CADUCA, pero hay gente que también dice expira) y otra cuando la contraseña EXPIRA (generalmente se dice EXPIRA pero hay gente que también dice caduca)

- La EXPIRACION de la contraseña se usa cuando desea obligar a alguien a renovar su contraseña en un intervalo específico de tiempo (tal vez 30 días) pero desea que la cuenta permanezca habilitada. Se le pedirá al usuario que cambie su contraseña en el próximo inicio de sesión. La EXPIRACION de la contraseña significa que el usuario debe cambiar la contraseña en el próximo inicio de sesión. El usuario aún podrá iniciar sesión después de restablecer la contraseña.
- En el caso de CADUCIDAD de la cuenta, la cuenta de usuario se bloqueará después de ciertos días que uno especifique, por lo que NO podrá iniciar sesión en absoluto. La CADUCIDAD de la cuenta debe usarse en otro contexto. Por ejemplo, si contrata a un contratista por 6 meses y desea que la cuenta se deshabilite al final del contrato, puede establecer una CADUCIDAD de la cuenta en una fecha fija y no tendrá que preocuparse por eliminar el acceso a este contratista, ya que al finalizar el contrato (a los 6 meses) la cuenta se desactivará automáticamente.

Ejercicios:

1. Crear los siguientes usuarios y grupos para su servidor, luego comprobar que puede ingresar con cada uno de los usuarios desde un terminal o consola.

Sugerencias:

- Utilice la misma contraseña para todos para que ingresar rápidamente a cada usuario y hacer la verificación, solo por temas de tiempo.
- Para hacer todas estas configuraciones y/o creación de usuarios y grupos, tiene que estar como usuario root.
- Primero cree los grupos luego **cree los usuarios y asígneles al grupo** en cuestión utilizando un solo comando (esto se vio en pasos anteriores), de esta manera acorta el proceso de creación y asignación.
- Conserve las minúsculas y evite los acentos en los nombres, de tal manera que le queden los nombres igual como en la tabla.

Grupos	Usuarios
abastecimiento	mgaray, jhuaman, rrios
sistemas	jpantoja, lquispe, rzavala
administracion	vchachi, gvargas, jrodriguez
despacho	fsalazar, pizquierdo, jjimenez
cuentas	grandes, fsalgado, lpanta

2. El usuario "jpantoja" sale de vacaciones, deshabilite su cuenta. (Deshabilite su "login")
3. Cambié al usuario "lpanta" al grupo "sistemas"

Solución:

1.

Crearemos el grupo "abastecimiento" y luego creamos el usuario "mgaray", al cual añadiremos al grupo "abastecimiento" al momento de su creación, repita el mismo proceso para los demás usuarios.

Importante: Entre como root para poder crear grupos y usuarios.

Comando o Instrucción	Explicación o comentario
\$ sudo su	Entre como root, si se encuentra con un usuario común utilice "sudo su"
# groupadd abastecimiento	Creación del grupo "abastecimiento"
# adduser mgaray --ingroup abastecimiento --gecos "Usuario de abastecimiento"	
Por ejemplo, asigne contraseña: otiuni	

2.

Comando o Instrucción	Explicación o comentario
# vi /etc/passwd	Editamos el archivo
Cambie el Shell del usuario "jpantoja" a → usr/sbin/nologin	
:wq!	Guardar y salir
Realicé las verificaciones de que el "login" para "jpantoja" fue deshabilitado	

3.

Comando o Instrucción	Explicación o comentario
# id lpanta	Vemos los grupos primarios y los grupos secundarios a los cuales pertenece lpanta
# usermod -g sistemas lpanta	Cambiamos el grupo primario de lpanta, ahora su grupo primario sera "sistemas" # usermod -g nos sirve para
# id lpanta	Vemos los grupos primarios y los grupos secundarios a los cuales pertenece lpanta. Verificamos el cambio de grupo a "sistemas"

Ejercicio complementario al ejercicio 3 anterior

¿Profesor usted cambio de grupo primario en el ejercicio 3, también podemos añadir o cambiar grupos secundarios? Si, se puede. Vemos un ejemplo:

Recordemos que los grupos que creamos fueron:

abastecimiento, sistemas, administracion, despacho, cuentas

Probemos añadiendo al usuario "lpanta" a algunos grupos secundarios, recordemos que de lo realizado anteriormente el usuario "lpanta" ahora tiene como grupo primario a "sistemas", entonces ahora por ejemplo digamos que queremos agregar al usuario "lpanta" a los grupos secundarios "abastecimiento" y "administracion" y luego queremos cambiarlo a los grupos secundarios "despacho" y "cuentas"

Importante: Entre como root para poder crear grupos y usuarios.

Añadir uno o más grupos secundario al usuario

Recordemos que podemos añadir un usuario hasta 15 grupos secundarios

Sintaxis:

usermod -a -G GROUP1, GROUP2,...,GROUPN USER

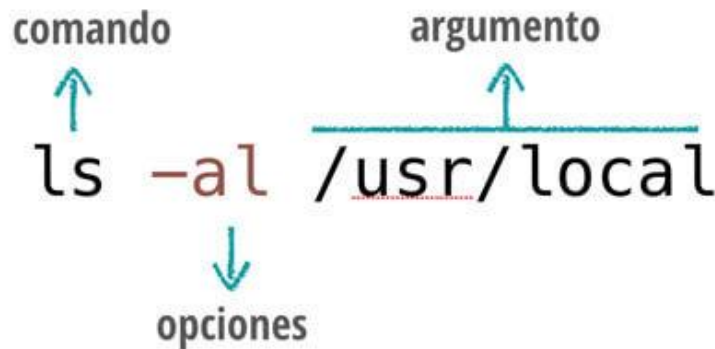
Comando o Instrucción	Explicación o comentario
# usermod -a -G abastecimiento,administracion lpanta	Agregamos al usuario "lpanta" a los grupos secundarios "abastecimiento" y "administracion"
# id lpanta	Verificamos

Debe obtener el siguiente resultado:

```
root@linux-oti-uni:/home# id lpanta
uid=1002(lpanta) gid=1003(sistemas) groups=1003(sistemas),1004(abastecimiento),1005(administracion)
root@linux-oti-uni:/home#
```

Asociado al "gid" esta el grupo primario del usuario lpanta, en este caso su grupo primario será "sistemas", luego en el campo groups nos aparece primero el grupo primario "sistemas" y a continuación nos aparecerán los grupos secundarios que en este caso son "abastecimiento" y "administracion"

Recordar las partes de un comando:



Anatomía del comando `ls` que lista las carpetas y archivos (en este caso de la carpeta `/usr/local`). La opción `-a` muestra los archivos ocultos y `-l` muestra los detalles de cada carpeta/archivo.

Todos los comandos se componen de:

- Un nombre con el que se invoca el comando.
- Opciones que modifican el comportamiento del comando. Son opcionales.
- Argumentos sobre los que actúa el comando. También opcionales.

Reemplazar lista de grupos secundarios de un usuario

Sintaxis:

`usermod -G GROUP1, GROUP2,...,GROUPN USER`

Acá a diferencia de lo anterior donde usamos la opción `a` y `G`, **solo usaremos la opción `G`**. Veamos el porque:

Comando o Instrucción	Explicación o comentario
<code># usermod -G despacho,cuentas lpanta</code>	Reemplazamos la lista de grupos secundarios antigua, donde estaban los grupos <code>abastecimiento</code> y <code>administracion</code> a una nueva lista de grupos secundarios donde estarán los grupos <code>despacho</code> , <code>cuentas</code> . Es decir, los nuevos grupos secundarios del usuario <code>lpanta</code> serán <code>despacho</code> y <code>cuentas</code> .
<code># id lpanta</code>	Verificamos

Debe obtener el siguiente resultado:

```
root@linux-oti-uni:/home# id lpanta
uid=1002(lpanta) gid=1003(sistemas) groups=1003(sistemas),1002(cuentas),1006(despacho)
root@linux-oti-uni:/home#
```

Asociado al “gid” esta el grupo primario del usuario lpanta, en este caso su grupo primario será “sistemas”, luego en el campo groups nos aparece primero el grupo primario “sistemas” y a continuación nos aparecerán los grupos secundarios que en este caso son “cuentas” y “despacho”

Si nosotros hubieramos usado la opcion “-a” al comando y hubiéramos ejecutado el comando:

usermod -a -G despacho,cuentas lpanta

Este hubiera añadido los grupos "despacho" y "cuentas" a la lista de grupos secundarios pero no la hubiera reemplazado, es decir la lista de grupos secundarios hubiera incluido a los grupos abastecimiento,administracion,**despacho,cuentas**

Conclusión: Vemos que el primero (opcion -G) reemplaza totalmente y el segundo (opciones -a -G) solo añade, para este caso REEMPLAZAREMOS la lista completa de grupos secundarios a "despacho" y "cuentas"

Eliminar o quitar a un usuario de uno o más grupos secundarios:

Precisiones:

- No confundir, una cosa es eliminar un grupo y otra es eliminar a un usuario de un grupo.
- Solo podemos eliminar o quitar a un usuario de uno o más grupos secundarios, pero no podemos eliminar a un usuario de su grupo primario, ya que obligatoriamente debe tener un grupo primario, lo que podemos hacer es cambiar de grupo primario al usuario.

Ahora lpanta tiene como grupos secundarios a despacho y cuentas, probemos eliminando a lpanta del grupo secundario "despacho".

Para remover un usuario de un grupo secundario necesitas sobrescribir los grupos a los que actualmente este usuario pertenece, por una nueva lista de grupos dónde no esté el grupo que se quiere remover.

Es decir, vamos a "REEMPLAZAR" la lista completa grupos secundarios a una lista donde no esté el grupo que se quiere remover, esto lo haríamos haciendo:

usermod -G cuentas lpanta

De esta manera hemos reemplazado la antigua lista de grupos secundarios que tenía a los grupos despacho y cuentas a una nueva lista de grupos secundarios donde solo esta lpanta.

Verificamos:

id lpanta

```
root@linux-oti-uni:/home# id lpanta
uid=1002(lpanta) gid=1003(sistemas) groups=1003(sistemas),1002(cuentas)
root@linux-oti-uni:/home#
```

Vemos que efectivamente se “removió” o “elimino” al usuario lpanta del grupo “despacho”

Cambio de Fecha y Hora del sistema:

Para cambiar fecha y hora, debemos entrar como usuario “root”.

Comando o Instrucción	Explicación o comentario
# timedatectl set-ntp 0	Con este comando desactivamos la sincronización automática de fecha y hora, que por defecto esta activada. Ubuntu 22.04 utiliza el demonio del servidor NTP (Protocolo de tiempo de red), que se conecta a un grupo de servidores de Protocolo de tiempo de red para obtener actualizaciones de tiempo constantes y precisas.
# date --set "25 Nov 2024 13:15:00"	Fijamos la fecha al 25 de Noviembre de 2024 con hora igual a 13 horas con 15 minutos y 00 segundos
# hwclock --systohc	Ahora establecemos el reloj de hardware en el BIOS del sistema, para que la configuración persista durante un reinicio. Durante el inicio del sistema, la hora del sistema se establece en el valor del reloj de hardware).
# date	Verificamos el cambio
# reboot	
# date	El cambio persiste después un reinicio.
# timedatectl set-ntp 1	Para revertir este cambio, activamos la sincronización automática.
# hwclock --systohc	Ahora establecemos el reloj de hardware en el BIOS del sistema.
# dpkg-reconfigure tzdata	Elegimos nuestra zona horaria. America/Lima en nuestro caso.
# date	Verificamos y efectivamente, probemos ante un reinicio a continuación.
# reboot	La fecha y hora sincronizada automáticamente se asigna correctamente. Se sugiere que deje su servidor con esta sincronización automática.

¿Qué es un daemon (demonio) en los entornos de UNIX o GNU/Linux?

Un servicio, programa residente o daemon (terminología de Unix, traducido erróneamente como demonio) es un tipo especial de programa que se ejecuta en segundo plano, en vez de ser controlado directamente por el usuario. Este tipo de programas continúa en el sistema, es decir, que puede ejecutarse de forma persistente o reiniciarse si se intenta matar el proceso dependiendo de la configuración del daemon y de las políticas del sistema. El uso de este nombre viene del retroacrónimo "Daemon", Disk And Execution Monitor (Monitor de disco y ejecución).

¿Cómo podemos ver los usuarios y grupos en una sola salida?

Para eso estando como usuario normal o root, ejecute la siguiente secuencia de programación:

```
# for user in $(awk -F: '{print $1}' /etc/passwd); do groups $user; done
```

Si quiere obtener las ultimas **N** líneas coloca:

```
# for user in $(awk -F: '{print $1}' /etc/passwd); do groups $user; done | tail -N
```

Si quiere obtener las primeras **N** líneas coloca:

```
# for user in $(awk -F: '{print $1}' /etc/passwd); do groups $user; done | head -N
```

Si desea saber que grupos son primarios y secundarios para un usuario en específico utilice:

```
# id NOMBREDEUSUARIO
```

Recordar: Una forma de ver rápidamente cual es el "grupo primario" y los "grupos secundarios" de un usuario es ingresando el comando "id NOMBREDEUSUARIO". A la salida de este comando nos saldrá el gid asociado al grupo primario (el gid caracteriza al grupo primario de un usuario) y en groups aparecerá el grupo primario (con su gid) seguido del grupo o de los grupos secundarios a los cuales pertenece el usuario.

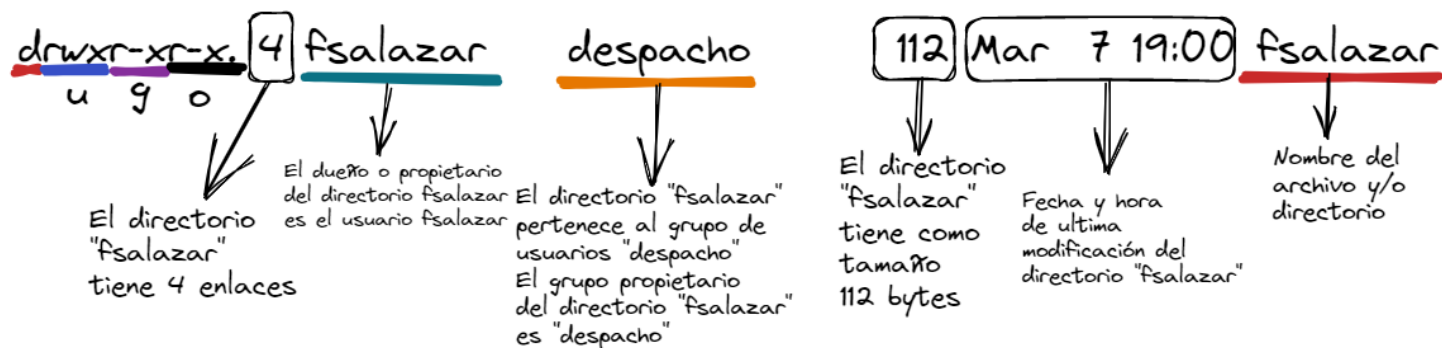
Explicación detallada del comando "ls -l"

Por ejemplo, sea:

```
$ sudo su
    Entramos como root
# cd /home
# ls -l
total 0
drwxr-xr-x 4 fsalazar  despacho    112 Mar  7 19:00 fsalazar
drwx----- 2 fsalgado  cuentas      62 Mar  6 10:15 fsalgado
drwx----- 4 grandes   cuentas      91 Mar  6 10:26 grandes
drwx----- 2 gvargas   administracion 62 Mar  6 10:13 gvargas

# Analicemos la primera linea de la salida del comando ls -l
drwx----- 4 fsalazar  despacho    112 Mar  7 19:00 fsalazar
```

La información que obtenemos es la siguiente:



fsalazar es un directorio (d), donde el dueño de este directorio es el usuario "fsalazar" y el grupo al que pertenece este directorio es "despacho".

El dueño del directorio "fsalazar" que es el usuario "fsalazar", tiene los permisos de r (lectura), w (escritura) y x (ejecución).

Los usuarios que pertenecen al grupo que pertenece el directorio "fsalazar", que es el grupo "despacho" tienen los permisos de r (lectura) y x (ejecución).

Los usuarios OTROS se refiere a los usuarios que pertenecen a los grupos secundarios a los cuales también puede pertenecer el usuario dueño o a otros usuarios ajenos con los que el usuario dueño no comparte ningún grupo (primario o secundario), estos OTROS tienen los permisos de r (lectura) y x (ejecución).

Importante:

Cuando un archivo o directorio se crea, de forma predeterminada tendrá como grupo propietario al grupo "primario" del usuario que lo creó (usuario dueño o propietario).

Grupo principal: especifica un grupo que el sistema operativo asigna a los archivos creados por el usuario. Cada usuario debe pertenecer a un grupo primario.

Grupos secundarios: especifica uno o más grupos a los que también pertenece un usuario. Los usuarios pueden pertenecer a hasta 15 grupos secundarios.

Nota: Un grupo de usuarios o grupo puede ser primario o secundario. Un usuario puede tener un grupo primario y hasta 15 grupos secundarios.

Introducción a los permisos en el manejo de grupos

Al entrar con el usuario **jhuaman** que pertenece al grupo "**abastecimiento**" (al cual pertenecen los usuarios **mgaray**, **jhuaman**, **rrios**) quisiéramos entrar al directorio de nuestros compañeros que están en el mismo grupo, por ejemplo:

Comando o Instrucción	Explicación o comentario
Entrar el usuario jeanleonardo	Entramos con el usuario "jeanleonardo"
\$ cd /home/rrios	Con el usuario "jeanleonardo" vamos al directorio hogar del usuario rrios y nos saldrá acceso denegado -bash: cd: /home/rrios: Permission denied
\$ cd /home/mgaray	-bash: cd: /home/mgaray: Permission denied

¿En ambos casos sale que tenemos el "permiso denegado", pero porque ocurre esto? Esto ocurre porque para los directorios hogares de los usuarios rrios y mgaray, el usuario jeanleonardo cae en la categoría de "otros", porque no es el usuario dueño de los directorios ni pertenece a ninguno de los grupos propietarios de los directorios, y podemos observar que el campo "otros"

tiene los permisos “ - - - ”. Es decir los usuarios y/o grupos que caen en la categoría de otros no tienen permisos de lectura (r) , escritura (w) y ejecución (x). Por eso nos sale el mensaje de permiso denegado en ambos casos.

```

jeanleonardo@linux-oti-uni:/home$ ls -l
total 68
drwxr-x--- 2 fsalazar      despacho      4096 mar 16 00:58 fsalazar
drwxr-x--- 2 fsalgado      cuentas       4096 mar 16 00:59 fsalgado
drwxr-x--- 2 grandes       cuentas       4096 mar 16 00:58 grandes
drwxr-x--- 2 gvargas       administracion 4096 mar 16 00:57 gvargas
drwxr-x--- 4 jeanleonardo  jeanleonardo 4096 mar 16 14:52 jeanleonardo
drwxr-x--- 2 jhuaman       abastecimiento 4096 mar 16 00:54 jhuaman
drwxr-x--- 2 jjimenez      despacho      4096 mar 16 00:58 jjimenez
drwxr-x--- 6 jneyra        almacen       4096 mar 15 20:47 jneyra
drwxr-x--- 2 jpantoja      sistemas      4096 mar 16 00:55 jpantoja
drwxr-x--- 2 jrodriguez     administracion 4096 mar 16 00:57 jrodriguez
drwxr-x--- 2 lpanta        sistemas      4096 mar 15 22:02 lpanta
drwxr-x--- 2 lquispe       sistemas      4096 mar 16 00:55 lquispe
drwxr-x--- 2 mgaray        abastecimiento 4096 mar 16 00:54 mgaray
drwxr-x--- 2 pizquierdo    despacho      4096 mar 16 00:58 pizquierdo
drwxr-x--- 2 rrios         abastecimiento 4096 mar 16 00:54 rrios
drwxr-x--- 2 rzavala       sistemas      4096 mar 16 00:56 rzavala
drwxr-x--- 2 vchachi       administracion 4096 mar 16 00:56 vchachi

```

Que podemos hacer para que el usuario “jeanleonardo” puede ver el contenido de los directorios hogar de los usuarios “rrios” y “mgaray”. Lo que tenemos que hacer es entrar con cada usuario, “rrios” y “mgaray”, ir al directorio /home y desde ahí cambiar los permisos asignados al campo “otros” para cada directorio hogar de los usuarios en cuestion, de tal manera que asignemos el permiso de lectura (r) al campo “otros”, con esto realizado el usuario “jeanleonardo” podrá ver el contenido de los directorios hogares de los usuarios “rrios” y “mgaray”

Comando o Instrucción	Explicación o comentario
Entrar el usuario rrios desde una terminal.	Entramos con el usuario “rrios”
cd /home	
chmod o+rx rrios/	Asignamos permisos de lectura y ejecución al campo “otros”
Le debe quedar lo siguiente: drwxr-xr-x 3 rrios abastecimiento 4096 mar 16 16:06 rrios Nota: chmod viene de CHANGE MODE (cambiar modo)	
Entrar el usuario jeanleonardo desde una terminal.	
cd /home/rrios	
ls	

Nota: Repita lo mismo con el usuario “mgaray”

¿Qué hacen realmente los permisos en un archivo de Linux?

Read (r): El permiso de lectura se utiliza para acceder al contenido del archivo. Puede usar una herramienta como `cat` o `less` el archivo para mostrar el contenido del archivo. También puede usar un editor de texto como `Vi` o `view` el archivo para mostrar el contenido del archivo. Se requiere permiso de lectura para hacer copias de un archivo, porque necesita acceder al contenido del archivo para hacer un duplicado.

Write (w): El permiso de escritura le permite modificar o cambiar el contenido de un archivo. El permiso de escritura también le permite usar los operadores de redirección o agregar en el shell (`>` o `>>`) para cambiar el contenido de un archivo. Sin permiso de escritura, no se permiten cambios en el contenido del archivo.

Execute (x): El permiso de ejecución le permite ejecutar el contenido de un archivo. Por lo general, los ejecutables serían cosas como comandos o aplicaciones binarias compiladas. Sin embargo, el permiso de ejecución también permite que alguien ejecute scripts de shell de Bash, programas de Python y una variedad de lenguajes interpretados.

¿Qué hacen realmente los permisos en un directorio de Linux?

Los tipos de archivos de directorio se indican con **d**. Conceptualmente, los permisos funcionan de la misma manera, pero los directorios interpretan estas operaciones de manera diferente.

Read (r): Al igual que los archivos normales, **este permiso le permite leer el contenido del directorio**. Sin embargo, eso **significa que puede ver los contenidos (o archivos) almacenados en el directorio**. Este permiso es necesario para que funcionen cosas como el comando `ls`.

Write (w): Al igual que con los archivos normales, **esto permite que alguien modifique el contenido del directorio**. Cuando está cambiando el contenido del directorio, está agregando archivos al directorio o eliminando archivos del directorio. Como tal, debe tener permiso de escritura en un directorio para mover (`mv`) o eliminar (`rm`) archivos de él. También necesita permiso de escritura para crear nuevos archivos (usando `touch` o un operador de redirección de archivos) o copiar (`cp`) archivos en el directorio.

Execute (x): Este permiso es muy diferente en los directorios en comparación con los archivos. Esencialmente, puede pensar que proporciona acceso al directorio. **Tener permiso de ejecución en un directorio le autoriza a mirar información extendida sobre archivos en el directorio (usando `ls -l`, por ejemplo) pero también le permite cambiar su directorio de trabajo (usando `cd`) o pasar a través de este directorio en su camino a un subdirectorio debajo.**

¿Puede crear archivos o directorios dentro del directorio hogar de “rrios” y “mgaray” con el usuario “jeanleonardo” después de haber realizado la modificación de permisos?

```
jeanleonardo@linux-oti-uni:/home/rrios$ mkdir ejemplo
mkdir: cannot create directory 'ejemplo': Permission denied
jeanleonardo@linux-oti-uni:/home/rrios$ touch ejemplo
touch: cannot touch 'ejemplo': Permission denied
jeanleonardo@linux-oti-uni:/home/rrios$
```

Por ejemplo, ingrese con “jeanleonardo” vaya al directorio hogar de “rrios”, e intente crear un directorio y archivo de nombre ejemplo, el resultado que obtendrá será al de la imagen mostrada anteriormente.

Y como podemos hacer que el usuario “jeanleonardo” si pueda crear archivos y directorios dentro de los directorios hogar de “rrios” y “mgaray”? Simple, asigne el permiso de escritura (w) al campo otros (o), veamos:

Comando o Instrucción	Explicación o comentario
Entrar el usuario rrios desde una terminal.	Entramos con el usuario “rrios”
cd /home	
chmod o+w rrios/	Asignamos permisos de escritura al campo “otros”
Le debe quedar lo siguiente: drwxr-xrwx 3 rrios abastecimiento 4096 mar 16 16:06 rrios	
Entrar el usuario jeanleonardo desde una terminal.	
cd /home/rrios	
mkdir ejemplo	
touch ejemplo.txt	¿Por qué no le asigna el nombre “ejemplo”, al igual que al directorio? Hágalo y vea que pasa. Usted notará que el archivo (al hacer touch ejemplo) no se creará. ¿Por qué? Vea el apartado “Importante” más abajo para ver la explicación de esto.
ls -l	
rm -rf *	

Nota: Repita lo mismo con el usuario “mgaray”

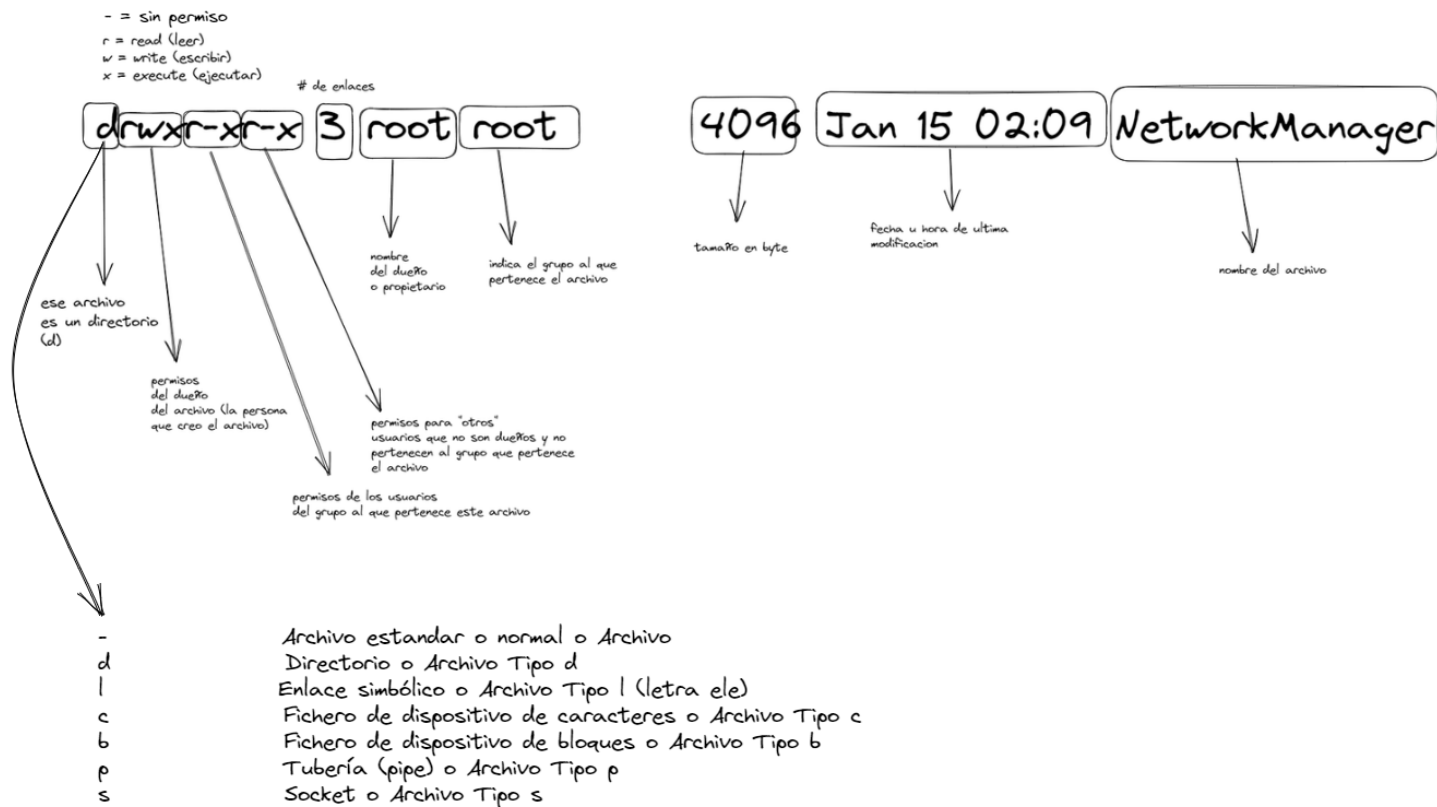
Importante:

En Linux “todo” es archivo. Al ingresar el comando “ls -l” el primer carácter de la izquierda nos indica el tipo de archivo.

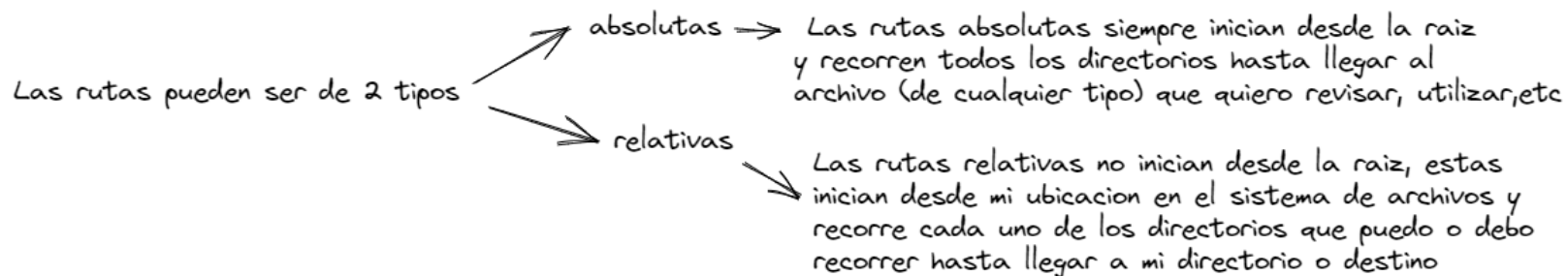
Carácter	Tipo
-	Archivo estandar o normal o Archivo
d	Directorio o Archivo Tipo d
l	Enlace simbólico o Archivo Tipo l (letra ele)
c	Fichero de dispositivo de caracteres o Archivo Tipo c
b	Fichero de dispositivo de bloques o Archivo Tipo b
p	Tubería (pipe) o Archivo Tipo p
s	Socket o Archivo Tipo s

Por lo tanto, dentro de un directorio no puede haber un directorio (archivo tipo d) y un archivo normal que tengan el mismo nombre, porque como todo es archivo, no puede haber dos “archivos” con el mismo nombre.

Identificando permisos y tipo de archivo a través del comando "ls -l"



Rutas absolutas y relativas



Ejemplo de ruta absoluta:

Indicador de ruta absoluta

↓ Empieza desde la raíz

/etc/ssh/sshd-config.d/

Ejemplo de ruta relativa:

Por ejemplo supongamos que nos encontramos en /home
Tenemos el directorio hogar del usuario "carlos" que dentro tiene
los directorios "ventas" "deudas". Una ruta relativa sería:

carlos/ventas/
↑
Indicador de ruta relativa
NO empieza desde la raiz

UMASK (Mascara del usuario)

- **Permite establecer permisos** a los ficheros u archivos **por defecto** esto afecta **a la creación de archivos, directorios, enlaces, etc.**
- Una máscara se compone de 3 dígitos octales, en base a la cual se calculan los permisos finales por defecto que tendrán los archivos y/o directorios creados por un usuario.

En binario	---->	Numero decimal	--->	Numero octal
*****		*****		*****
000		0		0
001		1		1
010		2		2
011		3		3
100		4		4
101		5		5
110		6		6
111		7		7

- Un numero binario de tres dígitos puede representar un numero o dígito octal (dígitos del 0 al 7)
- Tomamos la base octal porque únicamente va de 0 a 7.
- El valor de la máscara lo podemos cambiar o modificar, ya veremos después como hacerlo.

Si asignamos un valor binario a cada permiso r, w y x tenemos:

r = 1
w = 1
x = 1
- = 0

Entonces tomemos como ejemplo la siguiente cadena de permisos:

`rwxr-xr-x = 111 101 101 (en binario) = 7 5 5 (octal tomando de 3 en 3)`