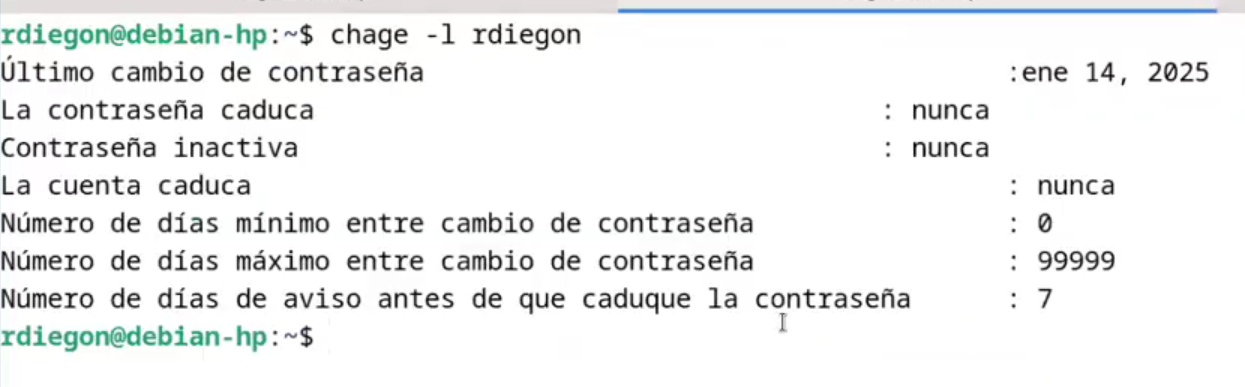
Clase 11

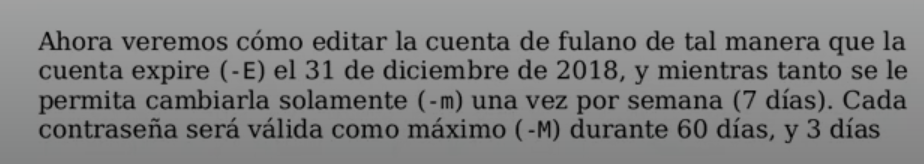
Clase de permisos / permisos especiales

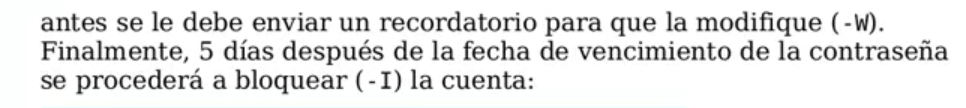
// Command: chage - l <user>



Se usa para editar los datos que salen en la captura

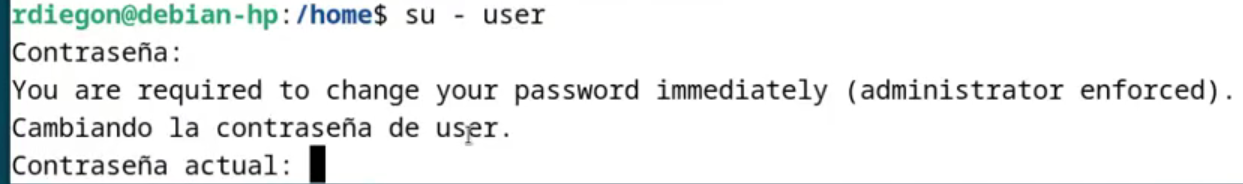
Por ejemplo, con chage -E 2025-12-31 -m 7 -M 60 -W 3 -I 5 fulano  
De esta manera, expira con -E en esa fecha



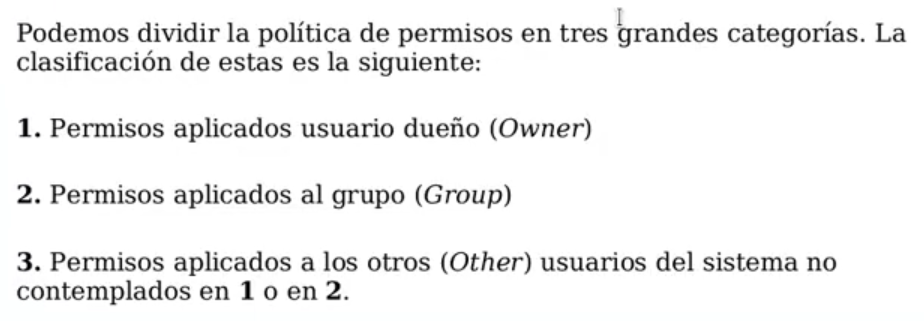


Tambien se usa para que el usuario, se loguea por primera vez tenga que cambiar su passwd

chage -d 0



Permisos

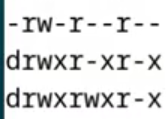


Tenemos permisos para ellos, que son de READ (r) - WRITE (w) - EXECUTE (x)

r: lectura / comandos de lectura como ls,cat,etc

w:escritura / le permitira editar archivos de texto por ejemplo, todo lo que tenga que ver con editar algo o modificarlo

x: ejecución / permita la ejecución de scripts por ejemplo, en carpetas funciona para aperturas de carpetas alias acceder



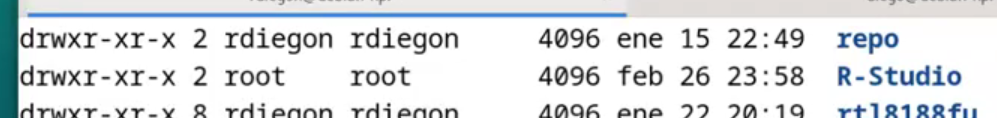
file - owner: read/write/none - group: read/none/none - others: read/none/none

directory - owner: read/write/excute - group: read/none/execute - others: read/none/execute

//Command: sudo

Si creo o ejecuto cosas con sudo, se hacen con el usuario ROOT a pesar de estar de estar logueado con otro usuario

Ejemplo:



//Command: chown <user>:<group> <recurso>



Si esta asi, se modifica el recurso para el usuario rdiegon  
Si esta como sudo chown :rdiegon solo para el grupo rdiegon

Si esta en ambos lados, se cambia para los 2

A su vez, se puede usar alternativamente solo grupos con chgrp

//Command: chmod <a quien>

Existen 2 metodos, el octal y el nominal para cambiar los permisos del recurso

<a quien> : u -> user / g -> groups / o -> others / a -> todos

Eso es para el user owner y groups owner

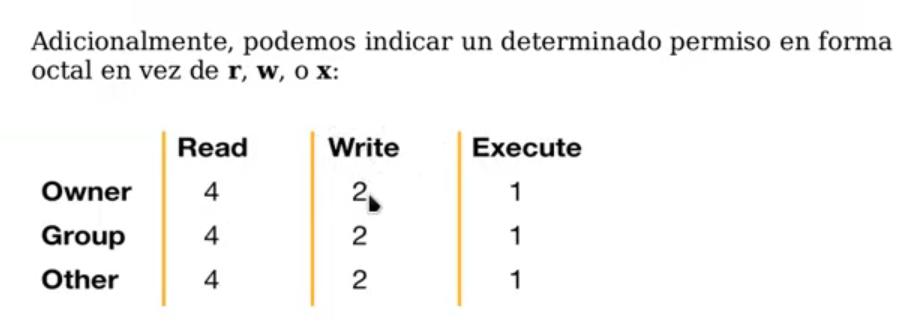
Se le quita con menos y se le suma con mas

sudo chmod go+w prueba

Se le suman los permisos de write a prueba para los grupos owner y others

IMPORTANTE MANEJAR LOS PERMISOS DE MANERA EFICIENTE

Los permisos de AUTOEXECUTE no se presuponen, por ende tiene de read/write



chmod 4+2+1|4+2|4+2 <file> -> chmod 766 prueba

# Mascaras

Permisos por defecto que va a tener un archivo nuevo

[\\Command](file:///\\Command): umask

Te devuelve los permisos que va a tener el archivo nuevo/recién creado

Ejemplo: touch file1.txt

Devuelve umask -> Ejemplo: 022

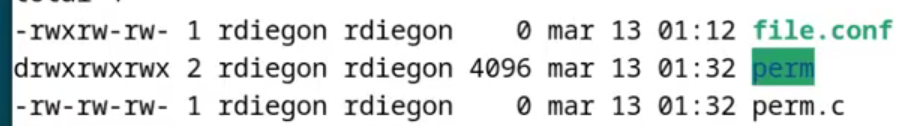
Ejemplo: 

Permisos tope definidos por umask: 644 + umask = 666

Permisos tope definidos por umask: 755 + umask = 777

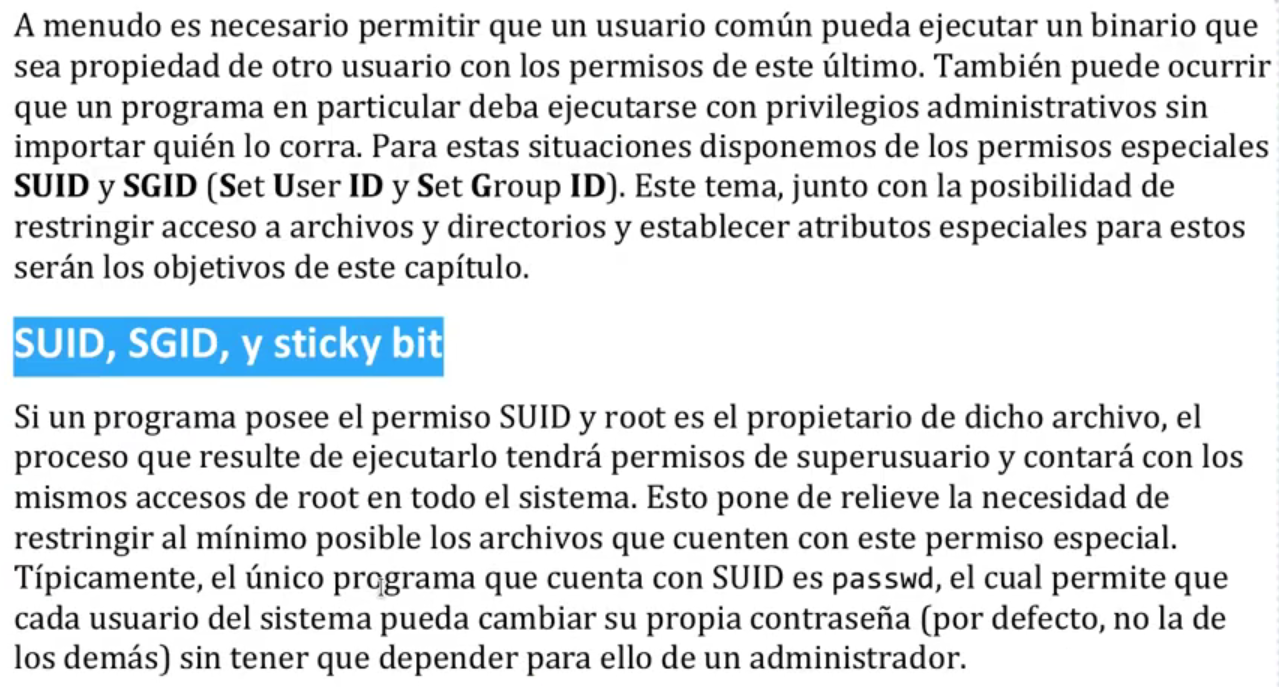
Por ende, los archivos tiene un 666 como máximo  
Y la mascara solo “resta” en formato octal los permisos que podrían llegar a tener para files/directory nuevos

Ejemplo: umask = 000 - newfile: 666

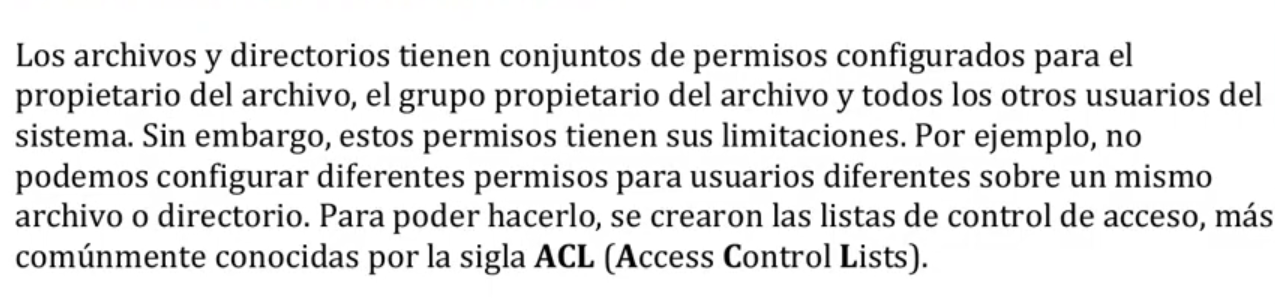


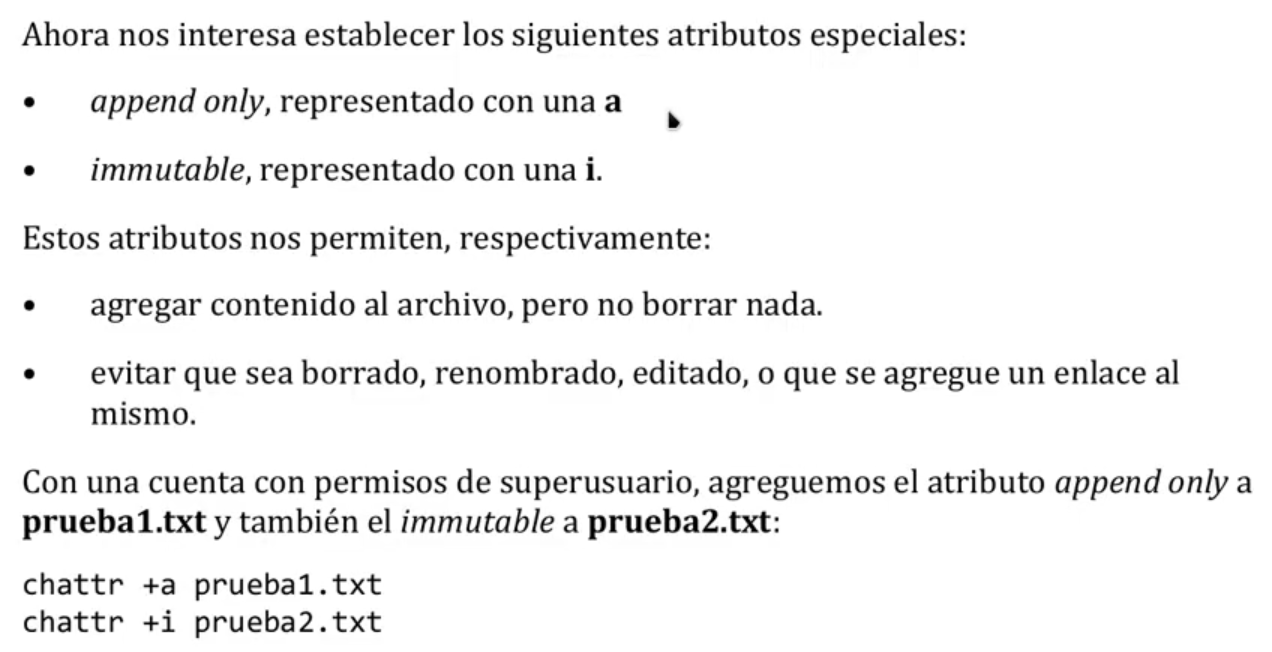
Se puede modificar la mascara para el usuario logueado y se aplica lo que se cree a partir de ese cambio, no modifica el resto

# Permisos Especiales



# ACL (Access Control Lists)





# Sticky Bit

Permisos 777 generalmente, pero la única cosa que no se puede hacer es ELIMINAR  
Solo el owner