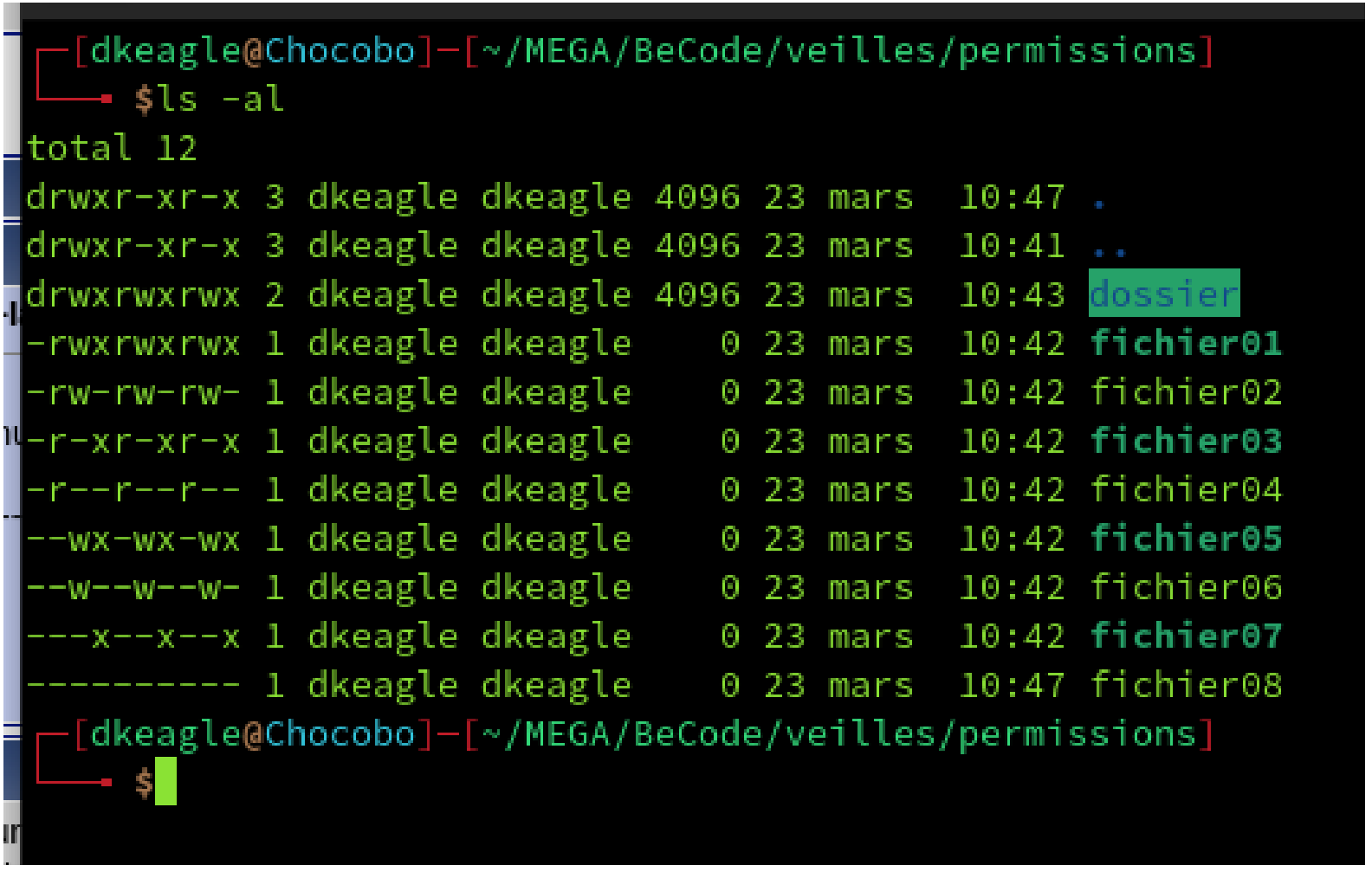
Les droits Linux
Commandes et permissions

La gestion des droits sous Linux se dit DAC (pour Discretionary Access Control => Contrôle d’accès discretionnaire)

Les droits d’un fichier ou d’un dossier sont divisés en trois groupes : ceux du propriétaire (user), ceux du groupe propriétaire (group) et ceux des autres (others).

Chacun de ces groupes est lui aussi divisé en 3 bits, le premier concerne la lecture, le second l’écriture et le dernier, l’exécution.

/!\ Le premier bit de permission ne représente en réalité pas une permission, mais indique s’il s’agit d’un fichier ‘-’ ou d’un dossier ‘d’


[dkeagle@Chocobo]-[~/MEGA/BeCode/veilles/permissions]
$ls -al
total 12
drwxr-xr-x 3 dkeagle dkeagle 4096 23 mars 10:47 .
drwxr-xr-x 3 dkeagle dkeagle 4096 23 mars 10:41 ..
drwxrwxrwx 2 dkeagle dkeagle 4096 23 mars 10:43 dossier
-rwxrwxrwx 1 dkeagle dkeagle 0 23 mars 10:42 fichier01
-rw-rw-rw- 1 dkeagle dkeagle 0 23 mars 10:42 fichier02
-r-xr-xr-x 1 dkeagle dkeagle 0 23 mars 10:42 fichier03
-r--r--r-- 1 dkeagle dkeagle 0 23 mars 10:42 fichier04
--wx-wx-wx 1 dkeagle dkeagle 0 23 mars 10:42 fichier05
--w--w--w- 1 dkeagle dkeagle 0 23 mars 10:42 fichier06
---x--x--x 1 dkeagle dkeagle 0 23 mars 10:42 fichier07
---------- 1 dkeagle dkeagle 0 23 mars 10:47 fichier08
[dkeagle@Chocobo]-[~/MEGA/BeCode/veilles/permissions]
$

Changer ces permissions avec la commande «chmod»

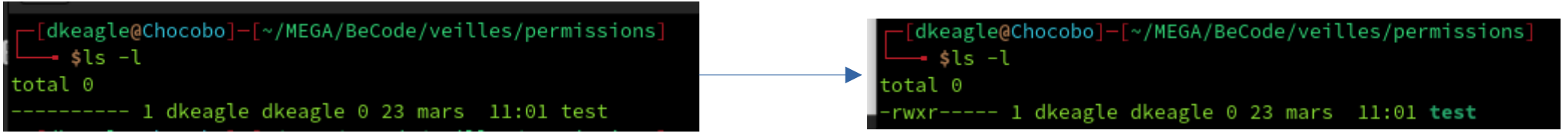
La commande chmod peut-être utilisée de beaucoup de façons, avec de multiples options et paramètres.

Les deux plus connues et utilisées sont la méthode numérique (chmod 777 *fichier*) et la méthode additive (ou soustractive) (chmod u+x *fichier*).

La méthode numérique fonctionne sur un simple calcul binaire afin de définir «quel droit» doit être associé à «quel groupe» :

Ainsi, chmod sera suivi de 3 valeurs octales (chiffre allant de 0 à 7)

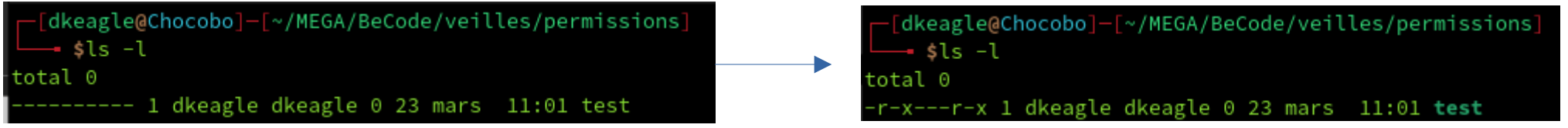
i.e : chmod 740 *fichier*
7 sera donc la permission accordée au premier groupe (propriétaire)
4 sera la permission accordée au second groupe (groupe propriétaire)
0 sera finalement la permission accordée au dernier groupe (les autres)



La méthode additive (ou soustractive) fonctionne sur une opération (+ ou -) entre un (ensemble de) groupe(s) et un (ensemble de) droit(s)

Ainsi, chmod sera suivie de lettres (pour les groupes), d'un symbole + ou – et d'autres lettres (pour les droits cette fois)

i.e : chmod uo+rx *fichier* ajoutera '+' les droits de lecture 'r' et exécution 'x' aux propriétaire 'u' et aux autres 'o'



Les droits spéciaux

Plus tôt, j’ai expliqué que les permissions étaient configurées selon 3 bits

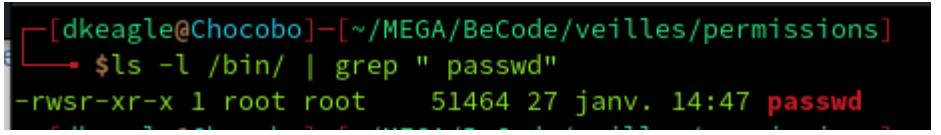
Eh bien j’ai menti.

Il y a en effet 2 autres bits.

le SetUID/SetGID bit et le Sticky bit

Le SetUID/SetGID (pour Set User ID & Set Group ID) remplace le bit d'execution 'x' et à un utilisateur d'exécuter un script sans pour autant en avoir le droit.

i.e : le script «passwd» (situé dans /bin/) permet à un utilisateur de changer de mot de passe.



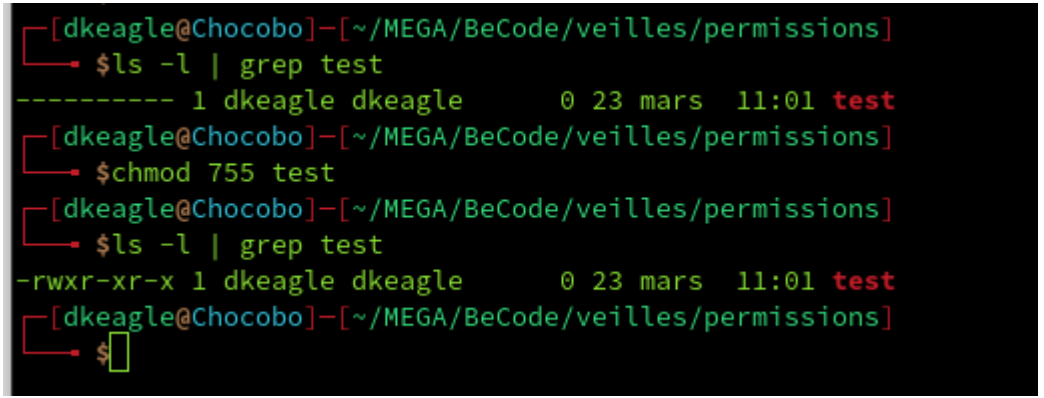
On peut voir que ce script appartient à root et au groupe root, et que le droit d'exécution du propriétaire est indiqué 's' et non 'x'.

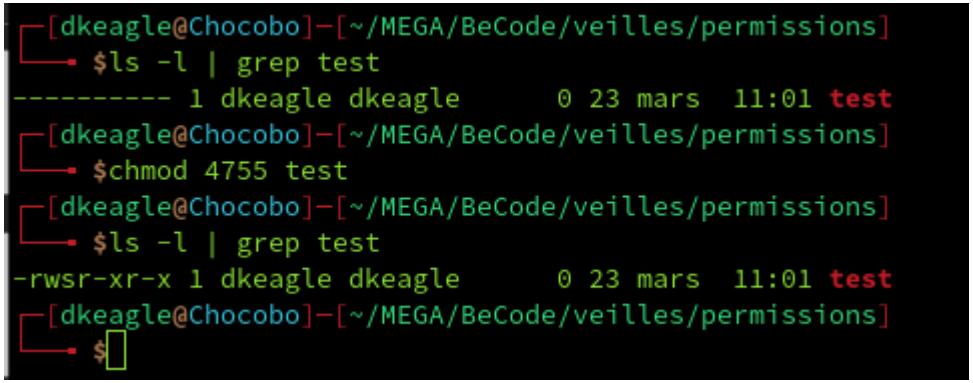
Cela signifie que quand un utilisateur lambda utilise le script /bin/passwd, il s'exécute en prenant l'User ID de son propriétaire (root) et non pas celui de l'utilisateur qui l'a exécuté.

En résumé, le SetUID/setGID bit permet d'exécuter un script en tant que propriétaire de celui-ci.

On peut appliquer le SetUID/SetGID bit à un fichier via la méthode numérique vue plus tôt, cette fois-ci en commençant la série de valeur par un 4.

Ainsi

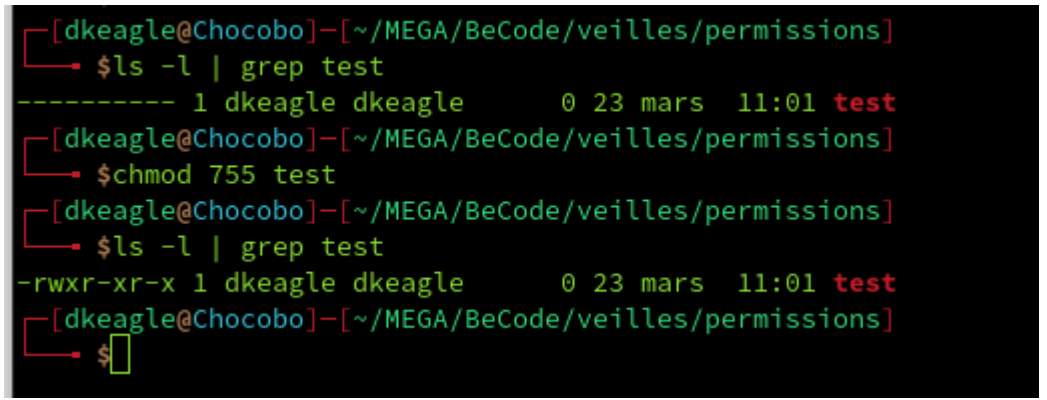


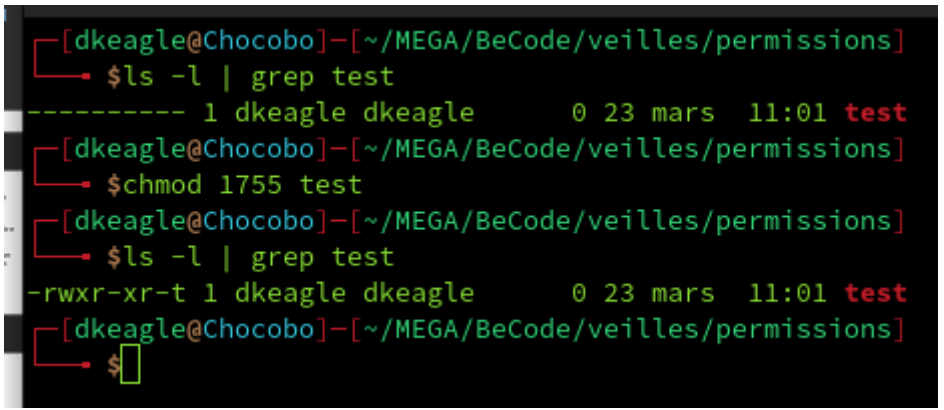


Le Sticky bit permet de partager un fichier avec beaucoup d'autres utilisateurs, sans pour autant leur donner le droit de supprimer ledit fichier

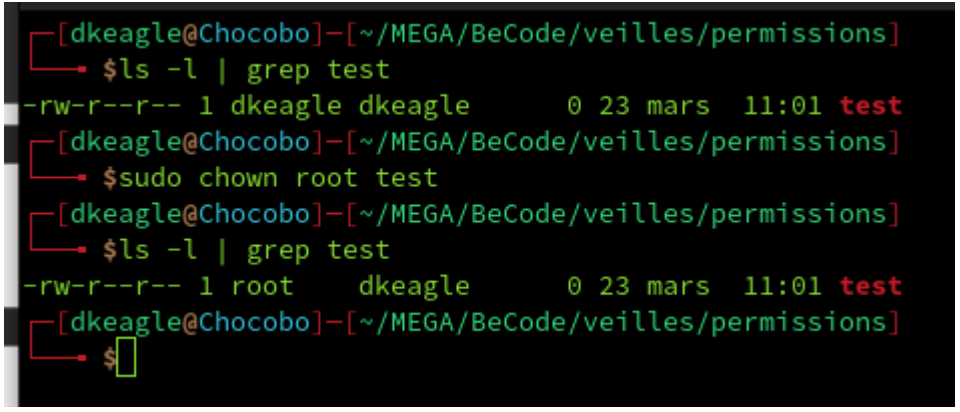
Il est représenté par un 't' remplaçant le droit d'exécution du groupe «autres» (others).

On l'applique comme le SetUID/SetGID bit, en ajoutant un 1 devant la valeur numérique (au lieu d'un 4)





Les commandes ‘chown’ (CHange OWNer) et ‘chgrp’ (CHange GRouP) permettent quant à elles de changer le propriétaire ou le groupe propriétaire d’un fichier.



/!\ chown et chgrp nécessitent une élévation de privilèges (sudo) pour être utilisées

/!\ Ces commandes sont à utiliser avec prudence !

Merci à tous !