La suite … des commandes usuelles *sh

Permissions ugo

Compression et archives tar.gz

Permissions (protection) des Fichiers (et Répertoires)

Pour chaque fichier, Unix distingue 3 classes d'utilisateurs : '**u**', '**g**' et '**o**' ('**a**' englobe les 3)

- Le propriétaire : **user**
- Les membres du groupe : **group**
- Le reste du monde : **other**

Pour chaque classe, Unix distingue aussi 3 types de permissions

- Lecture : '**r**' (comme **read**)
- Écriture : '**w**' (comme **write**)
- Exécution : '**x**' (comme **eXecute**, pour un répertoire cela permet de le traverser)

Ces permissions sont indépendantes les unes des autres

3 classes X 3 types = 9 permissions par fichier (d'autres attributs sont parfois ajoutés **s**, **t**)

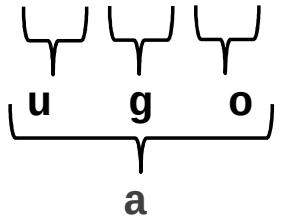
Chaque permission est vraie (donnée) ou fausse (retirée)

Affichage long ls -l ou stat

- permission absente

rwx : permission présente

```
$ cd /etc
syska@linserv1:/etc$ ls -l passwd
-rw-r--r-- 1 root root 1886 sept. 10 09:31 passwd
$ stat -c %a passwd
644
$ stat -c %A passwd
-rw-r--r--
```



Modes octal ou symbolique

chmod [OPTION]... MODE[,MODE]... FICHIER...

chmod [OPTION]... MODE-OCTAL FICHIER

Exemples :

- chmod go+x toto titi

  Ajouter (+) perm exécution (x) au groupe et aux autres (go) pour les fichiers toto et titi
- chmod a-rx tutu

  Retirer (-) les permissions lecture et exécution (rx) à toutes les catégories (a) pour le fichier tutu
- chmod u=rx titi

  Fixer exactement (=) les permissions pour le propriétaire (u)du fichier titi à rx
- chmod 644 titi

Mode octal : table de correspondance

Binaire	Logique	Octal
000	(---)	0
001	(--x)	1
010	(-w-)	2
011	(-wx)	3
100	(r--)	4
101	(r-x)	5
110	(rw-)	6
111	(rwx)	7

Pourquoi s'embêter avec des permissions en octal ?

Seules certaines combinaisons sont usuelles :

Répertoires : 755, 750 ou 700

- 7 : rwx pour le propriétaire
- 5 : rx pour les autres (et/ou le groupe)
- 0 : Aucun droit pour les autres (et/ou le groupe)

  Rappel : x est indispensable pour « traverser » un répertoire.

Fichiers : 644, 640 ou 600 (755 pour une commande)

Certaines commandes ne comprennent que l'octal :

`umask` : retrait systématique de certaines permissions

Ex : tester **`umask 002`** puis **`touch titi`**

find : recherche (profonde) de fichiers

Ex : **`find ./ -perm +100`**

Fichiers dont la permission vaut au moins 100 = exécutables pour le propriétaire

tar: manipulation d’archives

Regrouper en un seul fichier – archive -plusieurs autres fichiers ou répertoires

Préserver les arborescences associées ainsi que les propriétés des fichiers : permissions, dates, propriétaire, codage des caractères des noms de fichiers ....

"Mise à plat" des répertoires : format Ustar

Exemple : tar cvf ~/Racine.tar Racine

Format couramment utilisé (Java jar, zip, …)

Permet de diffuser / stocker des arborescences sur des FS aux capacités parfois limitées (*fat)

tar : commande simple

tar cvf MonArchive.tar Repertoire_1

tar cvf MonArchive.tar Repertoire_1 Repertoire_2 …

tar xvf MonArchive.tar

tar xvf MonArchive.tar -C Repertoire_Destination

Options z (ou j)

Une des options suivantes doit être utilisée :

–A, ––catenate, ––concatenate Met bout–à–bout plusieurs fichiers d'archive.

–c, ––create Crée une nouvelle archive.

–d, ––diff, ––compare Cherche les différences entre les fichiers indiqués et ceux contenus dans le fichier archive.

––delete Supprime les fichiers indiqués du contenu d'une archive. Cette option ne doit pas être utilisée avec les lecteurs de bandes !

–r, ––append Sauvegarde les fichiers indiqués à la fin d'une archive.

–t, ––list Affiche la liste des fichiers contenus dans une archive.

–u, ––update Ne sauvegarde que les fichiers plus récents que ceux de même nom déjà présents dans l'archive.

–x, ––extract, ––get Restaure les fichiers contenus dans une archive.

gzip: compression de fichiers

Réduire la taille d'un fichier

Algorithme de compression sans perte (contrairement à JPEG pour les photos)

Exemples :

gzip Racine.tar

produit le fichier Racine.tar.gz, ou directement:

tar zcvf ~/Racine.tar.gz Racine

tar : options plus complexes

Append (ajouter des fichiers dans une archive) :

```
gunzip TP_Linux.tar.gz
tar rvf ~/TP_Linux.tar /SupportCours/S1T/M111
gzip TP_Linux.tar.gz
```

option exec de la commande find

find OPTIONS [CHEMIN…] EXPRESSION ACTION

Pour chaque fichier trouvé, on applique le traitement donné après **-exec** terminé par **\;**

Le nom du fichier trouvé est **{}**

Exemple :

```
find /SupportCours/S1T –name '*.pdf' –o –name '*.zip' –exec cp {} ~/VRAC \;
```

```
find / \( -name tmp -o -name '*.xx' \) -atime +7 -exec rm {} \;
```

Efface tous les fichiers dont le nom est tmp ou qui finissent par .xx et qui n'ont pas été accedes depuis 7 jours

option exec de la commande find

SI on trouve les fichiers suivants avec **find**

$ find . -name '*.tar'

toto.tar

tmp/titi.tar

tmp/toto.tar

ALORS la commande

$ find . -name '*.tar' -exec gzip {} \;

est équivalente à la suite de commandes donnée à droite

$ gzip toto.tar

$ gzip tmp/titi.tar

$ gzip tmp/toto.tar

Traduction en langage courant : pour tous les fichiers trouvés en dessous / à partir du répertoire courant dont le nom termine par la chaîne .tar, appliquer la commande gzip au fichier.