Université Mohammed V de Rabat
Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes
Année Universitaire 2021/2022

Professeur : **Abdellatif Kobbane**

TP 7 Administration Unix : Droits d'accès

1 Les permissions de base:

Les permissions de base pour lire (r), écrire (w), ou exécuter (x) peuvent être associées à chaque fichier et répertoire dans le système de fichiers UNIX. La signification de ces permissions pour les fichiers et les répertoires sont:

- Pour les fichiers:
  - **Read (r) =** permet l'accès pour lire le contenu du fichier et le copier.
  - **Write (w)** = permet de modifier le contenu du fichier
  - **Execute (x) =** permet l'exécution du fichier s'il est binaire ou script shell.
- Pour les répertoires :
  - **Read (r)** = permet de lister le contenu du répertoire.
  - **Write (w) =** permet l'ajout et la suppression des fichiers et des répertoires dans ce répertoire.
  - **Execute (x)** = permet la recherche, l'accès et la traversé d'un répertoire pour atteindre des fichiers et répertoire qu'il contient

2 Classes d'accès

Les permissions s'appliquent à trois classes d'accès qui sont:

- **User (u)** = qui représente le propriétaire du fichier.
- **Group (g)** = groupe propriétaire du fichier.
- **Other (o)** = tous les autres utilisateurs du système qui ne sont ni propriétaire ni dans le groupe propriétaire..

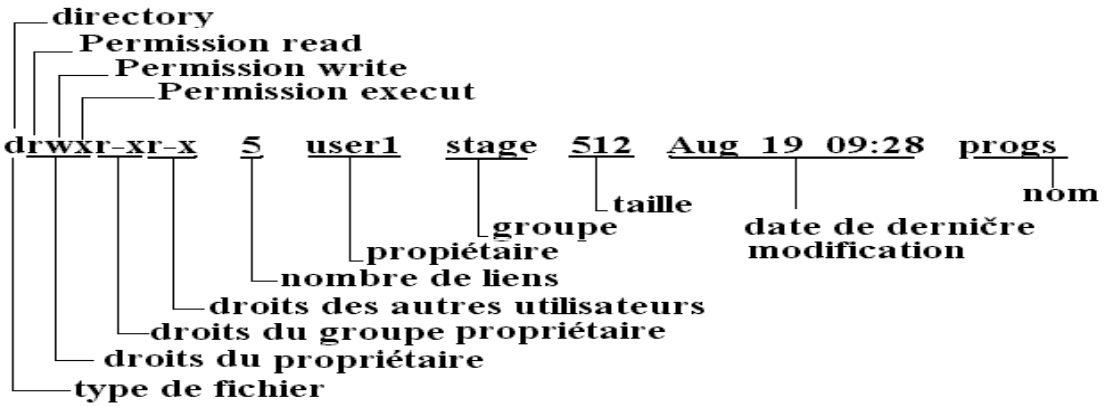
il faut noté que chaque fichier dans le système UNIX est la propriété d'un utilisateur et d'un groupe. Donc si on prend un utilisateur quelconque et un fichier quelconque dans le système, alors cette utilisateur est soit le propriétaire du fichier, soit il est membre du groupe propriétaire sinon il est considérer comme faisant parti des autres utilisateurs

3 Affichage des droits d'accès : ls -l

Pour afficher les attributs des fichiers et répertoires, on utilise l'option -l de la commande ls tel que dans la sortie suivante:

```
$ ls  -l
total  3
drwxr-xr-x   5   user1   stage     512   Aug 19  09:28    progs
-rw-r--r--   1   user1   stage     430   Aug 19  09:25    file1
-rwxr-xr-x   1   user1   stage     24    Aug 19  09:29    monscript
```

L'interprétation de ces différentes informations est donnée dans la figure suivante



4 Changement des permissions

Le propriétaire d'un fichier ou répertoire a la possibilité de changer les droits d'accès positionnés en utilisant la commande **chmod.** Deux syntaxes sont possibles:

- En mode absolu
- en mode symbolique

4.1 chmod en mode absolu :

La syntaxe de la commande chmod en mode absolu consiste à passer comme argument le mode souhaité et le(s) du (des) fichier(s)/répertoire(s) concerné(s) par l'opération.

```
$ chmod mode Nom-fichier
```

où mode est donné en octal, il se calcul en se basant sur les puissances de 2 (4, 2 et 1) et en prenant comme coefficient 1, si le droit correspondant existe, sinon 0. Par exemple :

User	Groupe	Other
r w x	r – x	r - -
1 1 1	1 0 1	1 0 0
1 x 4 + 1 x2 + 1 x 1	1 x 4 + 0 x2 + 1 x 1	1 x 4 + 0 x2 + 0 x 1
7	5	4

```
$chmod 754 fichier
```

Exemples : 775 indique rwx rwx r-x

640 indique rw- r- - - - -

1.1 chmod en mode symbolique :

Dans ce cas, la commande chmod nécessite la précision de la classes d'accès concernée par l'opération, le type de l'opération: ajouter ou supprimer le droit, et le droit concerné. Par exemple si on souhaite ajouter (+) de droit d'écriture (w) pour le groupe propriétaire sur un fichier f1 on utilise la commande

chmod g+w f1

d'une manière générale, la syntaxe est:

```
$ chmod classe opération permission fichier
```

CLASSE	OPERATION	PERMISSION
u : user (propriétaire) **g** : groupe propriétaire **o** : others (public) **a** : (all)	**-** supprimer le droit **+** ajouter le droit **=** fixer les droits	**r :** read **w** : write **x** : execute

Exemples:

```
$ls -l test
-rwxr-xr-x   1   user1   gp1   430   Aug 19  09:25   test
$chmod   o-r   test
$ls -l test
-rwxr-x--x   1   user1   gp1   430   Aug 19  09:25   test
$chmod   a-x   test
-rw-r-----  1   user1   gp1    430   Aug 19  09:25   test
$chmod   g=xw   test
-rw--wx---  1   user1   gp1    430   Aug 19  09:25   test
$chmod   =r   test
-r--r--r--  1   user1   gp1   430   Aug 19  09:25     test
```

2 La commande umask.

A la création d'un fichier ou d'un répertoire, le système positionne les valeurs de permissions par défaut. Ces valeurs sont gérées par la variable UMASK, qu'il est possible de changer par la commande umask. La valeur de UMASK (en octal) indique les droits qui doivent être masqués sur les répertoires et fichiers créés. La valeur de 1 indique que le droit correspondant doit être masqué et la valeur 0 indique qu'il doit apparaître.

Par exemple: si la valeur de UMASK est 022 en octal, ce qui correspond à 000 010 010 en binaire, alors les droits sur les répertoires seront : rwx r-x r-x. Ici le droit w pour le groupe propriétaire et autres sont masqués car dans le vecteur binaire de l'UMASK, leurs positions contient la valeur 1.

- Les droits obtenus sur un répertoire sont égale au complémentaire de la valeur de l'UMASK à 777 en octale
- Les droits obtenus sur un fichier sont égale à ceux d'un répertoire mais en plus le droit d'exécution (x) est aussi masqué.

Il est possible de changer les valeurs par défaut, pour la session courante, par la commande umask:

```
$umask code
```

Le changement de la valeur de l'UMASK n'affecte pas les droits des répertoires et fichiers déjà créés, mais elle s'appliquera sur les nouveaux répertoires et fichiers créés après le changement.

Exemple:

sur la sortie suivante, la valeur de l'UMASK est 002 ce qui donne les droits rwxrwxr-x sur les répertoires créés, et rw-rw-r-- sur le fichier. Par la suite, la valeur de l'umask est changé vers 027 ce qui donnera sur les répertoire créés aprs ce changement les droits rwxr-x---.

```
[user1@adrar tp3]$ umask
0002
[user1@adrar tp3]$
```

```
[user1@adrar tp3]$ mkdir tp33
[user1@adrar tp3]$ touch f33
[user1@adrar tp3]$ ls -l
total 12
-rw-rw-r-- 1 user1 gp1    0 mai  9 22:32 f33
drwxrwxr-x 2 user1 gp1 4096 mai  9 22:32 tp33
[user1@adrar tp3]$ umask 027
[user1@adrar tp3]$ mkdir tp44
[user1@adrar tp3]$ touch f44
[user1@adrar tp3]$ ls -l
total 16
-rw-rw-r-- 1 user1 gp1    0 mai  9 22:32 f33
-rw-r----- 1 user1 gp1    0 mai  9 22:33 f44
drwxrwxr-x 2 user1 pg1 4096 mai  9 22:32 tp33
drwxr-x--- 2 user1 gp1 4096 mai  9 22:33 tp44
```

Exercices

Exercice 1:

En étant connecté en tant qu'administrateur, mettez les droits **700** à l'ensemble des fichiers de l'utilisateur **user01**.

Exercice 2:

- En utilisant le manuel, retrouvez les différentes utilisations du droit **SGID** dans le système Linux.
- Créez un fichier par la commande **cp** et rendez-le non modifiable. Listez ses attributs. Essayez de le modifier.

Exercice 3:

La commande **`ls -l`** dans un répertoire affiche la sortie suivante :

```
[ab@adrar examen]$ ls -l
total 28
-rw-rw-r-- 2 ab exp   67 jan 18 13:49 app1.c
-rwsrwxr-x 1 ab exp 4838 jan 18 13:50 app1.o
-rw-rw-r-- 2 ab exp   67 jan 18 13:49 application
drwxrwxr-x 2 ab exp 4096 jan 18 13:45 cpp
-rwSrw-r-- 1 ab exp   67 jan 18 13:51 dev
drwxrwxr-x 2 ab exp 4096 jan 18 13:45 java
lrwxrwxrwx 1 ab exp    9 jan 18 13:46 prog -> prog.java
-rw-rw-r-- 1 ab exp  240 jan 18 13:46 prog.java
```

1. Qui est le propriétaire de « app1.c»?
2. Qui est le groupe propriétaire de «cpp»?
3. Quelle est la nature de fichiers suivants: «app1.c », « cpp », « dev » et « prog »?
4. Peut-on dire que nous avons des liens durs dans cette sortie ? Si oui lesquels ?

5. Peut-on dire que nous avons des liens symboliques dans cette sortie ? Si oui lesquels?
6. Quels sont les droits du propriétaire, du groupe propriétaire et des autres sur « prog.java »?
7. Donner une seule commande pour supprimer les droits lecture et d'écriture aux autres utilisateur sur le fichier « app1.c »?
8. Donner une seule commande pour supprimer la lecture et l'écriture aux groupe propriétaire et autres sur « prog.java » ?
9. Quels sont les droits positionnés sur « app1.o » après lancement de la commande suivantes : **chmod 511 app1.o** ?
10. Que signifie les droits **s** (en minuscule) sur « app1.o » ci-dessus ? Et le droit S (en majuscule) sur « dev »?
11. Déduire la valeur de l'UMASK de cette utilisateur, sachant que uniquement les droits des fichiers qui contiennent s ou S ont été modifier?
12. Quels sont les droits obtenus sur un répertoire sachant que la valeur de UMASK est 266 ?

Exercice 4

Nous avons lancé la commande **ls -l** dans un répertoire, elle affiche la sortie suivante :

```
bash-2.04$ ls -l
total 20
drwxrwx---   2 bkp soft  4096 jun 14 14:46 simulations
drwxrwxr-x   2 bkp soft  4096 jun 14 14:45 travaux
lrwxrwxrwx   1 bkp soft     3 jun 14 14:46 modele -> modele2
-rw-rw-r--   2 bkp dev    509 jun 14 14:45 modele1
-rw-rw-r--   2 bkp dev    509 jun 14 14:45 modele4
-rw-------   1 bkp dev   4046 jun 14 14:44 modele2
-r-sr-xr-x   1 bkp dev   4046 jun 14 14:44 maj
```

1. Qui est le propriétaire de « **simulations** »?
2. Qui est le groupe propriétaire de « **simulations** »?
3. Quelle est la nature de **modele**? Pourquoi ? Donnez la commande qui nous a permis de créer **modele**?
4. Expliquer pourquoi les lignes des deux fichiers **modele1** et **modele4** sont elles identiques ? Donnez la commande qui nous a permis de créer **modele4**?
5. Quels sont les droits du propriétaire, du groupe propriétaire et des autres sur le répertoire **travaux** ?
6. Donner une seule commande pour ajouter le droit de lecture au groupe propriétaire et lui supprimer le droit d'écriture sur le fichier **modele2**?
7. Donner une commande pour supprimer le droit de lecture et d'exécution au autres sur **travaux**?
8. Quels sont les droits positionnés sur **modele1** après lancement de la commande suivantes : **chmod 540 modele1** ?
9. Que signifie les droits **s** sur le fichier **maj** ci-dessus ?
10. Un utilisateur de login **ali** tente de modifier le fichier **modele2**, Est ce qu'il peut le faire ? Pourquoi ?
11. Supposons que le programme **maj** permet la modification de **modele2**, Est ce que l'utilisateur de login **ali** pourra utiliser ce programme pour faire la modification de **modele2** ? Justifier votre réponse ?
12. Quel est la valeur de **umask** qui permet d'avoir les droits **rwxr-x---** sur chaque répertoire crée ? Quels seront dans ce cas les droits sur les fichiers ?