Université d'Oran 1 Ahmed Ben Bella
Faculté des Sciences Exactes & Appliquées
Département d'Informatique

Module Systèmes d'Exploitation I (L2/S4)

Fiche TP n°4 **Droits d'accès** **Année 2015/2016**

Linux est un système multi-utilisateur. Chaque utilisateur appartient au moins à un groupe. Les fichiers ont donc des permissions visant 3 types d'utilisateurs :

- du propriétaire du fichier
- du groupe
- des autres utilisateurs.

Pour chaque type de personnes visées, il y a trois types de droits :

- r : droit de lecture
- w : droit d'écriture
- x : droit d'exécution.

A ces types de droits on associe un bit, prenant : 0 n'a pas droit et 1 a droit
On a donc des triplets du genre 111 (par exemple) pour chacun des types d'utilisateurs.
Exemple : **111 100 101** .

- le propriétaire a les droits 111, c'est-à-dire lecture, écriture et exécution
- le groupe a les droits 100, c'est-à-dire le droit de lecture
- les autres ont les droits 101, c'est-à-dire les droits de lecture et d'exécution.

Position binaire	Valeur octale	Les droits	Commentaire
000	0	- - -	Aucun droits
001	1	- - x	Executable
010	2	- w -	Ecriture
011	3	- w x	Ecrire et executer
100	4	r - -	Lire
101	5	r - x	Lire et executer
110	6	r w -	Lire et ecrire
111	7	r w x	Lire ecrire et executer

Octal

Exemple.
J'ai un script *bash* que je veux être le seul à pouvoir modifier, mais que les personnes de mon groupe pourront lire. Et que tous pourront exécuter. Je devrais mettre les droits suivants :

Utilisateur	U	G	O
Droits d'accès	r w x	r - x	- - x
Position binaire	1 1 1	1 0 1	0 0 1
Octale	7	5	1

On va donc exécuter la commande suivante : `chmod 751 script.sh`

Symbolique

Avec la méthode octale, il me faudrait tout décomposer pour seulement supprimer un droit. Mais il existe la méthode symbolique. Elle est de type : chmod [ugoa][+-][rwx].
C'est l'une de ces lettres **u** (propriétaire du fichier), **g** (groupe), **o** (les autres), **a** (tout le monde = **u** + **g** + **o**), suivie de + ou - pour respectivement ajouter ou supprimer les permissions, et la forme symbolique des permissions est de la forme **r** (*read* : lecture), **w** (*write* : écriture), **x** (exécution).

Exemple, pour pouvoir modifier ce fichier texte qui nous appartient : `chmod u+w fich.odt`
Plusieurs droits symboliques en les séparant par des virgules :¨`chmod u+rw,g+r,o+r,a-x fich.odt`
On a ajouté les droits en lecture et en écriture au propriétaire, on ajoute les droits de lecture au groupe, les droits de lecture aux autres, et on enlève les droits d'exécution à tous. Seul le propriétaire du fichier ou le super-utilisateur **ROOT** peut modifier les droits sur les fichiers et répertoires.

Les droits du répertoire

- **r** : lister les fichiers
- **w** : ajouter ou de supprimer des fichiers
- **x** : empêche de rentrer dans le répertoire, et de le lister.

Ex1 : compléter

octal	droits	octal	droits
0	- - -	4	
1		5	r - x
2		6	
3		7	r w x

Ex2: tapez : ls -la > fichier
Chmod 777 fichier
Pouvez vous lire le fichier par cat. Pouvez vous le modifier ech >> fichier.
Même question chmod 666 fichier.
Mêm question chmod 077 fichier.

Ex3 : cut et sort
Soit le fichier villes.txt :

```
france : paris
vietnam : ho chi minh
italie : roma
france : bordeaux
vietnam : hanoi
inde : delhi
```

Essayez : **grep 'I' villes.txt** **grep ' :h' villes.txt** **grep ' ^i' villes.txt** **grep 'i$' villes.txt**

Cut sélectionne une colonne de données :
Cut -c 1-3 villes.txt cut -d: -f1 grep vietnam villes.txt | cut -d: -f2
Sort villes.txt sort -t: -k2 villes.txt

Ex 4 ; join

Créer un fichier villes-pays.txt (villes ordonnées par pays)
Créer un fichier continents-pays.txt (continents ordonnées par pays) à partir de continents.txt

```
europe : france
europe : italie
asie : vietnam
asie : chine
```

Join rapproche deux fichiers sur une clé commune. Les fichiers doivent être triés.
Join -t : -1 2 -2 1 continents-pays.txt villes-pays.txt.

***t : délimiteur de champs 0**
-1 2 clé du premier fichier = second champ
-2 1 clé du second fichier = premier champs

Ex5 : kill
kill termine un a processus. Il faut d'abord obtenir son pid avec ps -ef , et ensuite kill -9 to le terminer. .

```
ps -ef | grep vim
******    7243  7222  9 22:43 pts/2    00:00:00 vim
kill -9 7243
```

Les shells

De nombreux shells sont disponibles sous Unix/Linux,

- sh
- bash (Bourne again shell),
- CSH (C Shell),
- KSH (KORN Shell),
- TCSH
- ...

Ils

- jouent le même rôle,
- ont des syntaxes différentes,
- fournissent des fonctions prédéfinies différentes.

Shell script

Définition : Shell-script = fichier texte qui contient une suite de commandes.

Exemple : fichier "premier.sh"

```
1 #
2 # Mon premier essai
3 #
4 clear
5 echo -n "Nous sommes le "
6 date
7 echo "et c'est mon premier script"
```

Note : le caractère `#` indique un commentaire

1. utiliser un éditeur de textes pour écrire le script
   Remarque
   - le suffixe .sh est recommandé
2. le rendre exécutable
   - `chmod +x mon-fichier.sh`
   - `chmod 755 mon-fichier.sh`

Démonstration

1. on tape le script suivant

```
1 #!/bin/bash
2 #
3 # fichier premier.sh
4 #
5 clear
6 echo "les scripts, c'est facile"
```

2. exécution par

   `bash ./first.sh`

Recommandation : la première ligne commençant par `#!` indique quel *shell* il faut utiliser.

1. lancement par

   `./first.sh`

   ne marche pas, parce que le script n'est pas exécutable
2. Rendre le fichier exécutable

```
$ chmod +x first
$ ./first
```