

TP 2 : Droits d'accès et liens

1 DROITS D'ACCÈS DES FICHIERS

Étant donné qu'Unix est un système d'exploitation multi-utilisateurs, on distingue trois catégories d'utilisateurs :

- u : propriétaire (user : celui qui a créé le fichier).
- g : le groupe (les autres utilisateurs appartenant au même groupe du propriétaire).
- o : les autres (others).

Ainsi, chaque fichier possède trois types de droits :

- r : lecture (read) ;
- w : écriture (write) ;
- x : exécution (execute) ;
- - : aucun droit.

Au niveau répertoire, ces droits signifient :

- Droit de lister les fichiers présents dans ce répertoire (ls) ;
- Droit de créer ou de détruire un fichier se trouvant dans ce répertoire ;
- Droit d'accéder à ce répertoire (cd)

EXERCICE 1

1. Déterminez votre ou vos groupes d'appartenance (id).

Réponse : la commande `id` permet d'afficher l'UID (ID de l'utilisateur connecté), le GID (ID du groupe principal du propriétaire) et les autres groupes avec leurs GID respectifs.

2. Utilisez `ls -l` depuis votre répertoire personnel, et repérez les symboles décrivant les droits, le propriétaire et le groupe d'appartenance d'un fichier de votre choix.

Réponse :

```
-rw-r--r-- 1 compte1 compte1 429 Dec 30 16:26 test.txt  
drwxr-xr-x 1 compte1 compte1 4096 Dec 30 20:06 toto
```

Dans cet exemple, le fichier `test.txt` appartient à l'utilisateur `compte1` et au groupe `compte1`. Le propriétaire a le droit de lecture et de modification, le groupe et les autres ont seulement le droit de lecture sur ce fichier.

3. Donnez les équivalences symboliques des droits d'accès suivants : 744, 633, 755, 111, 722 et 700

Réponse :

```
744 --> rwxr--r--  
633 --> rw--wx-wx  
755 --> rwxr-xr-x  
111 --> --x--x--x  
722 --> rwx-w--w-  
700 --> rwx-----
```

1. Créez un répertoire test et un fichier essai dans ce répertoire, et écrivez-y la phrase de votre choix.

Réponse :

```
mkdir test
cat > test/essai
Bonjour tout le monde
Ctrl+D
```

2. Notez à l'aide de `ls -l` les permissions actuelles du répertoire test et du fichier essai.

Réponse :

```
comptel@localhost$ ls -ld test
drwxr-xr-x 1 comptel comptel 4096 Dec 30 20:17 test
comptel@localhost$ ls -l test/essai
-rw-r--r-- 1 comptel comptel 22 Dec 30 20:17 test/essai
```

3. En utilisant la commande `chmod`, retirez-vous le droit en lecture et en écriture sur le fichier essai. Vérifiez l'effet obtenu en essayant d'afficher le contenu du fichier sur la fenêtre du terminal, puis de remplacer ce contenu par une phrase différente.

Réponse :

```
comptel@localhost$ chmod u-rw test/essai
comptel@localhost$ cat test/essai
cat: test/essai: Permission denied
comptel@localhost$ echo "Bonjour tout le monde" > test/essai
bash: test/essai: Permission denied
Nous n'avons pas le droit de visualiser le contenu de test/essai et nous n'avons pas de droit pour la modification du contenu.
```

4. Rétablissez le droit en écriture puis remplacez le contenu du fichier essai par le texte `echo "Ceci est un essai"`. Ajoutez-vous le droit en exécution, et exécutez le fichier essai en tapant `./essai` dans le terminal (depuis le répertoire qui le contient). Quel est le problème ?

Réponse :

```
comptel@localhost$ chmod u+w test/essai
comptel@localhost$ echo 'echo "Ceci est un essai"' > test/essai
comptel@localhost$ chmod u+x test/essai
comptel@localhost$ test/essai
bash: test/essai: Permission denied
En plus du droit d'exécution sur le fichier test/essai, il faut avoir le droit de lecture pour pouvoir l'exécuter
```

5. Rétablissez enfin le droit en lecture et tentez à nouveau d'exécuter le fichier.

Réponse :

```
comptel@localhost$ chmod u+r test/essai
comptel@localhost$ test/essai
Ceci est un essai
```

6. Placez-vous dans le répertoire test, et retirez-vous le droit en lecture pour ce répertoire. Listez le contenu du répertoire, puis affichez le contenu du fichier essai. Qu'en déduisez-vous ? Rétablissez le droit en lecture sur test.

Réponse :

```
comptel@localhost$ cd test
comptel@localhost$ chmod u-r .
comptel@localhost$ ls
ls: cannot open directory '.': Permission denied
comptel@localhost$ cat essai
echo "Ceci est un essai"
Nous ne pouvons pas lister le contenu du répertoire (il faut avoir le droit de lecture).
Par contre on peut toujours visualiser le contenu des fichiers du répertoire test,
tant que nous avons le droit de lecture sur ces fichiers.
```

7. Créez un fichier nommé nouveau dans test. Retirez à ce fichier et au répertoire test le droit en écriture. Tentez de modifier le fichier nouveau. Rétablissez ensuite le droit en écriture au répertoire test. Tentez de modifier le fichier nouveau, puis de le supprimer. Que constatez-vous ?

Réponse :

```
comptel@localhost$ touch nouveau
comptel@localhost$ chmod u-w . nouveau
comptel@localhost$ echo "cours des systèmes d'exploitation" > nouveau
bash: nouveau: Permission denied
comptel@localhost$ chmod u+w .
comptel@localhost$ echo "cours des systèmes d'exploitation" > nouveau
bash: nouveau: Permission denied
comptel@localhost$ chmod u+w .
comptel@localhost$ rm nouveau
rm: remove write-protected regular empty file 'nouveau'? y
Il suffit d'avoir le droit de modification sur le répertoire test pour pouvoir supprimer le fichier "nouveau".
```

8. Depuis votre répertoire personnel, retirez enfin le droit en exécution du répertoire test. Tentez de créer, supprimer, ou modifier un fichier dans le répertoire test, de vous y déplacer, d'en faire la liste, etc. Qu'en déduisez-vous quant au sens du droit en exécution pour les répertoires ?

Réponse :

```
comptel@localhost$ cd
comptel@localhost$ chmod u-x test
comptel@localhost$ touch test/nouveau
touch: cannot touch 'test/nouveau': Permission denied
comptel@localhost$ echo "Hello World" > test/essai
bash: test/essai: Permission denied
comptel@localhost$ cd test
bash: cd: test: Permission denied
```

On peut déduire qu'il faut avoir le droit d'exécution pour pouvoir accéder au répertoire, créer des fichiers ou de modifier le contenu des fichiers de ce répertoire.

EXERCICE 3 — UMASK

1. Donnez la commande pour connaître umask.

Réponse :

```
comptel@localhost$ umask
0002
```

2. Donnez la commande pour changer umask à 044.

Réponse :

```
comptel@localhost$ umask 044
```

3. Définissez un umask équilibré qui vous autorise un accès complet et autorise un accès en lecture aux membres de votre groupe. Testez sur un nouveau fichier et un nouveau répertoire.

Réponse :

On souhaite avoir les droits : rwx (7) pour le propriétaire, r--(4) pour le groupe et --- (0) pour les autres. Ainsi le masque à définir est le complémentaire à 777 de 740, ce qui donne $777-740=037$

4. Redémarrez votre console et vérifiez de nouveau la valeur de umask

Réponse :

Le masque sera rétabli à nouveau (022)

EXERCICE 3 - LIENS physiques ET LIENS symboliques

1. Faites deux copies du fichier /etc/passwd l'une appelée passwd1 et l'autre appelée passwd2.

Réponse :

```
comptel@localhost$ cp /etc/passwd passwd1
```

```
compte1@localhost$ cp /etc/passwd passwd2
```

2. Comparez leurs numéros d'inœuds à l'aide de la commande `ls -li`.

Réponse :

```
compte1@localhost$ ls -il passwd1 passwd2
1125899907344895 -rw-r--r-- 1 compte1 compte1 1991 Dec 30 21:30 passwd1
1125899907344896 -rw-r--r-- 1 compte1 compte1 1991 Dec 30 21:30 passwd2
```

3. Créez un lien physique à l'aide de la commande `ln` pour le fichier `passwd1` appelé `passwdph`.

Réponse :

```
compte1@localhost$ ln passwd1 passwdph
```

4. Comparez leurs numéros inœuds.

Réponse :

```
compte1@localhost$ ls -il passwd1 passwdph
1125899907344895 -rw-r--r-- 2 compte1 compte1 1991 Dec 30 21:30 passwd1
1125899907344895 -rw-r--r-- 2 compte1 compte1 1991 Dec 30 21:30 passwdph
```

5. Modifiez le contenu du fichier `passwd1`. Que remarquez-vous à propos du fichier `passwdph` ?

Réponse :

```
compte1@localhost$ echo "nouvelle phrase" > passwd1
compte1@localhost$ cat passwdph
nouvelle phrase
Les deux fichiers ont le même contenu
```

6. Créez un lien symbolique pour le fichier `passwd2` appelé `passwdsy` à l'aide de la commande `ln -s`.

Réponse :

```
compte1@localhost$ ln -s passwd2 passwdsy
```

7. Comparez leurs numéros d'inœus.

Réponse :

```
compte1@localhost$ ls -il passwd2 passwdsy
1125899907344896 -rw-r--r-- 1 compte1 compte1 1991 Dec 30 21:30 passwd2
6473924464853661 lrwxrwxrwx 1 compte1 compte1 7 Dec 30 21:36 passwdsy -> passwd2
```

8. Modifiez le contenu du fichier `passwd2`. Que remarquez-vous à propos du fichier `passwdsy` ?

Réponse :

```
compte1@localhost$ echo "abcd" > passwd2
compte1@localhost$ cat passwdsy
abcd
```

9. Supprimez le fichier `passwd2`. Que se passe-t-il pour le fichier `passwdsy` ?

Réponse :

```
compte1@localhost$ rm passwd2
le fichier passwdsy existe toujours mais ne pointe sur rien.
```

10. Editez de nouveau le fichier nommé `passwd2`. Que remarquez-vous ?

Réponse :

```
compte1@localhost$ cat > passwdsy
Bonjour tout le monde
Ctrl + D
compte1@localhost$ ls -l passwd2 passwdsy
-rw-r--r-- 1 compte1 compte1 22 Dec 30 21:44 passwd2
lrwxrwxrwx 1 compte1 compte1 7 Dec 30 21:36 passwdsy -> passwd2
On remarque la création du fichier passwd2 avec le contenu saisi par l'utilisateur.
```