

TP N° 4 : GESTION DES DROITS SOUS UNIX - CORRECTION

Pour ce TP, vous avez besoin d'être administrateur de votre machine afin d'être super-utilisateur et de pouvoir gérer les permissions sans restrictions.

I- Création d'utilisateurs et de groupes

Pour gérer les utilisateurs et les groupes on peut :

- soit manipuler directement les fichiers `/etc/passwd` et `/etc/group`
- soit utiliser un utilitaire graphique d'administration (par exemple `users-admin`)

Dans la suite du TP nous utiliserons la première solution

1. En modifiant le fichier `/etc/passwd`, créer l'utilisateur `user1`. Il aura un UID de 1005, un GID de 1005, un répertoire d'accueil nommé `/home/user1` et un shell `/bin/bash`. Le groupe de GID 1005 sera nommé `groupe12`.

Il faut ajouter dans `/etc/passwd` la ligne :

```
user1:x:1005:1005 ::/home/user1:/bin/bash
```

Il faut ajouter dans `/etc/group` la ligne :

```
groupe12:x:1005:user1
```

Dans une autre fenêtre, connectez-vous avec l'utilisateur `user1` (grâce à la commande `su - user1`). Si ça ne marche pas du premier coup, faites les modifications nécessaires pour que cela fonctionne (tester que cela fonctionne consiste à pouvoir se logger `user1` et créer un fichier)

- **premier problème** : le mot de passe. À la création de l'utilisateur `user1` dans `/etc/passwd`, aucun mot de passe n'est créé pour lui (allez voir le fichier `/etc/shadow` pour vous en convaincre). Il faut donc initialiser le mot de passe en tant qu'administrateur : tapez la commande : `sudo passwd user1`.
- **deuxième problème** : le répertoire d'accueil. À la création de l'utilisateur `user1` dans `/etc/passwd`, aucun répertoire d'accueil n'est créé pour lui (tapez la commande `ls /home` pour vous en convaincre). Il faut donc créer le répertoire d'accueil en tant qu'administrateur : tapez la commande : `mkdir /home/user1`.
- **troisième problème** : les droits du répertoire d'accueil. Le propriétaire et le groupe propriétaire de `/home/user1` sont `root` et `root`. Une fois connecté `user1` ne peut donc pas créer de fichier dans son répertoire d'accueil. Il faut mettre `user1` comme propriétaire de `/home/user1` et `groupe12` comme groupe propriétaire de `/home/user1`. Tapez les

commandes : `sudo chown user1 /home/user1` et `sudo chgrp groupe12 /home/user1`

2. Créer de même 2 utilisateurs : user2 et user3. Créer 3 groupes : groupe12 (avec user1 et user2) de GID 1005, groupe13 (avec user1 et user3) de GID 1006 et groupe23 (avec user2 et user3) de GID 1007. groupe12 sera le groupe principal de user1, groupe23 sera le groupe principal de user2, groupe13 sera le groupe principal de user3. Vérifier que les comptes sont fonctionnels.

- Il faut ajouter dans `/etc/passwd` les lignes :

```
user2:x:1006:1007 ::/home/user2:/bin/bash
```

```
user3:x:1006:1006 ::/home/user3:/bin/bash
```

- Il faut ajouter dans `/etc/group` les lignes :

```
groupe13:x:1006:user1,user3
```

```
groupe23:x:1007:user2,user3
```

et modifier la ligne :

```
groupe12:x:1005:user1,user2
```

- Il faut créer les mots de passe de user2 et user3 :

```
sudo passwd user2 ; sudo passwd user3
```

- Il faut créer les répertoires d'accueil de user2 et user3 :

```
sudo mkdir /home/user2 ; sudo mkdir /home/user3
```

- Il faut modifier les propriétaires des répertoires d'accueil de user2 et user3 :

```
sudo chown user2 /home/user2 ; sudo chgrp groupe23 /home/user2
```

```
sudo chown user3 /home/user3 ; sudo chgrp groupe13 /home/user3
```

3. Changer le shell de base de user3 : mettre un autre shell déjà installé sur la machine (ils sont dans le répertoire `/bin` et leur nom se termine par `sh`). Se connecter avec user1 et constater que le shell est différent.

`ls /bin/*sh` permet d'obtenir la liste des shell installés.

Modifier `/etc/passwd` comme ceci pour y mettre :

```
user1:x:1005:1005 ::/home/user1:/bin/sh
```

4. Changer le shell de base de user2 : mettre /bin/csh. Se connecter avec user2. Cela ne marche pas. Que faire pour résoudre le problème ? Remettre ensuite le shell bash.

Modifier /etc/passwd comme ceci pour y mettre :

```
user2:x:1006:1007 ::/home/user1:/bin/csh
```

Problème : en se connectant avec user2, /bin/csh n'existe pas. Il faut l'installer avec la commande `sudo apt-get install csh`.

Lorsqu'un utilisateur crée un fichier celui-ci lui appartient et le groupe propriétaire est son groupe principal.

1. En tant que user1, créer un fichier vide avec la commande `touch` (qui crée un fichier vide s'il n'existe pas déjà, et qui sinon change sa date de dernière modification).

```
touch vide
```

2. Avec la commande `ls -l`, quel est le propriétaire du fichier ? Le groupe propriétaire du fichier ?

le propriétaire du fichier = user1 Le groupe propriétaire du fichier = groupe12

3. Comment faire pour que le groupe propriétaire du fichier soit groupe13 plutôt que groupe12 ?

```
chgrp groupe13 vide
```

4. Faites pareil pour que le groupe propriétaire du fichier soit groupe23 plutôt que groupe13. Pourquoi cela ne marche-t-il pas ?

`chgrp groupe23 vide`. Cela ne marche pas car user1 ne fait pas partie du groupe groupe23.

Plusieurs commandes permettent d'avoir des informations sur l'utilisateur courant :

1. Taper les commandes `whoami` et `id`,

La commande `su` permet de changer son identité et de se connecter sous un autre identifiant

1. Taper la commande `su - user2`. Puis `whoami`. Quelle est la valeur de la variable d'environnement `HOME`? Faire `Ctrl^D` pour se déconnecter

II- Manipulation des droits d'accès aux fichiers

1- Le changement des droits : *chmod*

1. En tant que `user1`, créez un fichier `essai.txt` dans un répertoire `test`

```
mkdir test  
echo bonjour > test/essai.txt
```

2. Enlevez les droits de lecture du fichier `essai.txt` pour le propriétaire (`user1`).

```
chmod u-r test/essai.txt
```

Essayez ensuite de lire le fichier (avec la commande `cat` par exemple).

```
cat test/essai.txt
```

Remettez les droits de lecture.

```
chmod u+r test/essai.txt
```

3. Enlevez-vous les droits d'écriture au répertoire `test` qui contient `essai.txt`.

```
chmod u-w test
```

Vérifiez que vous ne pouvez plus créer de fichier dans le répertoire `test`.

```
touch test/tata
```

Essayez de supprimer le fichier `essai.txt`.

```
rm test/essai.txt
```

Remettez les droits d'écriture.

```
chmod u+w test
```

Enlevez-vous les droits d'exécution sur votre répertoire personnel.

```
chmod u-x test
```

Vérifiez que vous ne pouvez plus vous placer dedans.

```
cd test
```

4. Donnez le droit aux autres membres du groupe `groupe12` de venir lire la liste des fichiers contenus dans votre répertoire personnel.

```
chmod o-r test
```

Vérifiez en vous connectant en tant que `user2` puis `user3`

```
su - user2 ; cd ~user1/test ; ls :marche
```

```
su - user3 ; cd ~user1/test ; ls :ne marche pas
```

5. 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 ?

```
chmod u-x test
```

le droit x pour un répertoire empêche de le traverser

On peut aussi utiliser `chmod` en employant la notation octale pour désigner les permissions.

1. Remettez les droits d'exécution pour tous sur votre dossier personnel en utilisant `chmod` et la notation octale.

```
chmod 777 /home/user1
```

2. Toujours en utilisant la notation octale, enlevez le droit aux membres de votre groupe principal de lire le contenu de votre répertoire personnel. Testez en essayant d'y accéder en tant que `user2` ou `user3`.

```
chmod 737 /home/user1
```

```
su - user2 ; cd ~user1 ; ls : ne marche pas
```

```
su - user3 ; cd ~user1 ; ls : marche
```

2- Partage de répertoire :

Dans certains cas, par exemple dans le cadre d'un projet à plusieurs, il peut être intéressant de donner accès à certains de ses fichiers ou répertoires à d'autres utilisateurs.

Donner accès à ses fichiers en tant que `user1`.

1. Attribuez au fichier `essai.txt` les droits suffisants pour que `user2` (mais pas `user3`) puisse y accéder en lecture (mais pas en écriture).

```
chmod g+r essai.txt ; chmod o-r essai.txt
```

2. Vérifiez en vous connectant `user2` dans une autre console et `user3` dans une autre console

Ouvrir un répertoire d'accueil.

En tant que `user1` créez un fichier voisin dans votre répertoire `test` et fixez les droits nécessaires pour que :

- seul `user2` puisse écrire dans ce fichier,
- toute personne du groupe `groupe23` puisse lire ce fichier,
- vous seul puissiez effacer ce fichier ou créer de nouveaux fichiers dans `test`.

```
chown user2 test/voisin
```

```
chgrp groupe23 test/voisin
```

```
chmod 740 test/voisin
```

```
chmod 711 test
```

3- Modification du *PATH*

Connectez-vous comme `user1`. Créer un dossier `bin`. On souhaite y mettre des fichiers qui pourront être exécutés depuis n'importe où.

1. Créer un fichier `bonjour` qui contient la commande `echo bonjour`.

```
echo echo bonjour > bin/bonjour
```

2. Lancer le fichier `bonjour`. Cela ne marche pas, quel est le problème ?

```
bin/bonjour : cela ne marche pas; administrateur n'a pas le droit x sur bin/bonjour.
```

Comment y remédier ?

```
chmod u+x bin/bonjour
```

3. Modifiez ensuite la variable `PATH` pour que l'on puisse lancer la commande `bonjour` depuis n'importe où

```
PATH=$PATH:~user1/bin
```

```
bonjour
```

4. Modifier ensuite le fichier `.bashrc` de façon à ce que cette modification soit pérenne (si votre utilisateur n'a pas de `.bashrc`, copiez le depuis le répertoire d'accueil de l'administrateur ainsi que le fichier `.profile`). Testez.

Ajouter à la fin du fichier `.bashrc` la ligne :

```
export PATH=$PATH:/home/user1/bin
```

Connectez-vous comme `user3`.

1. Modifier le fichier `.bashrc` de `user3` et regardez si la modification fonctionne aussi. Quel est le problème ? Comment y remédier ?

```
user3 n'a pas le droit x sur /home/user1/bin/bonjour.
```

```
Pour l'ajouter user1 doit taper : chmod a+x /home/user1/bin/bonjour
```