

TP N° 4 : GESTION DES DROITS SOUS UNIX

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`. 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)
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.
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. Remettre ensuite le shell `bash`.
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 ?

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

3. Comment faire pour que le groupe propriétaire du fichier soit `groupe13` plutôt que `groupe12` ?
4. Faites pareil pour que le groupe propriétaire du fichier soit `groupe23` plutôt que `groupe13`. Pourquoi cela ne marche-t-il pas ?

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`
2. Enlevez les droits de lecture du fichier `essai.txt` pour le propriétaire (`user1`). Essayez ensuite de lire le fichier (avec la commande `cat` par exemple). Remettez les droits de lecture.
3. Vérifiez que vous ne pouvez plus créer de fichier dans le répertoire `test`. Essayez de supprimer le fichier `essai.txt`. Remettez les droits d'écriture. Enlevez-vous les droits d'exécution sur votre répertoire personnel. Vérifiez que vous ne pouvez plus vous placer dedans.
4. Donnez le droit aux autres membres du groupe `groupe12` de venir lire la liste des fichiers contenus dans votre répertoire personnel. Vérifiez en vous connectant en tant que `user2` puis `user3`
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 ?

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.
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`.

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).
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`.

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`
2. Lancer le fichier `bonjour`. Cela ne marche pas, quel est le problème ? Comment y remédier ?
3. Modifiez ensuite la variable `PATH` pour que l'on puisse lancer la commande `bonjour` depuis n'importe où
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.

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 ?