

## TP n°11 – Gestion des fichiers et des comptes

### 1) Liens

On va commencer par étudier les liens également appelés raccourcis dans Windows. Dans la machine LILI, créez un dossier appelé liens et entrez dedans.

#### a) Liens physiques

Créer un fichier appelé `recit.txt` et mettez n'importe quoi dedans : « salut ». Vérifiez avec `more` que ce fichier est bien créé et contient bien ce que vous avez mis.

Utilisez la commande `ln` pour créer un lien physique appelé `texte.txt`.

```
ln recit.txt texte.txt
```

Regardez le contenu de chacun de ces deux fichiers avec `more`. Éditez (avec `nano` ou `vi`) l'un puis éditez l'autre et à chaque fois vérifiez le contenu de chacun pour vous convaincre que c'est le même.

Attention, n'utilisez pas `emacs` car cet éditeur commence par renommer l'original en `original~` puis travaille sur une copie. Le lien serait donc perdu.

`ls -li` vous affiche les numéros d'i-nodes des fichiers : voyez que `recit.txt` et `texte.txt` ont le même numéro ce qui signifie qu'ils partagent le même contenu : c'est le même fichier mais il a deux noms.

On voit aussi le nombre de liens, juste avant `lili` administrateurs. Normalement pour `recit.txt` et `texte.txt`, ce nombre doit être de 2, tandis qu'il est à 1 pour un autre fichier.

Supprimez `recit.txt` et regardez ce qui arrive à `texte.txt`. (à rédiger dans le Wiki)

Recréez le lien dans l'autre sens :

```
ln texte.txt recit.txt
```

puis éditez à nouveau les deux en vérifiant le contenu de l'autre.

Supprimez `texte.txt` cette fois-ci et regardez ce qui arrive à `recit.txt`.

Si vous avez encore le fichier `tp10.iso`, remontez-le sur `/mnt/iso` à l'aide de `sudo mount -o loop tp10.iso /mnt/iso`

Essayez de créer un lien physique de `/mnt/iso/readme.txt` vers le dossier courant :

```
ln /mnt/iso/readme.txt readme.txt
```

On ne peut pas car un lien physique c'est un partage du même numéro d'i-node or ces numéros sont spécifiques du volume concerné. On pourrait le faire avec un lien logique.

#### b) Liens logiques

Faire la même chose avec des liens symboliques. La commande à employer est :

```
ln -s recit.txt texte.txt
```

Vérifiez avec `more` que les deux fichiers contiennent la même chose.

Constatez avec `ls -l` que pourtant les deux fichiers ne font pas la même taille : `texte.txt` fait seulement 9 octets, en fait c'est le nombre de caractères qu'il y a dans le nom « `recit.txt` ». En effet, un lien logique, c'est seulement un nom complet vers la cible.

Vous verrez que si vous supprimez `recit.txt`, alors `texte.txt` est « cassé », son contenu a disparu.

L'intérêt d'un lien logique (ou symbolique) est de pouvoir être créé sur un dossier et éventuellement entre des volumes différents.

Essayez de créer un lien logique de /mnt/iso/readme.txt vers le dossier courant :

```
ln -s /mnt/iso/readme.txt readme.txt
```

Ça marche... sauf si on démonte le volume /mnt/iso, le lien logique est cassé.

### **c) Étude des liens dans un filesystem ext4**

On part du répertoire liens dans lequel vous travaillez. Créez ceci :

```
mkdir -p d1/d2  
touch f1 d1/f2 d1/d2/f3  
tree  
ls -laiR
```

La commande `ls -laiR` affiche tous les numéros d'i-node de tous les dossiers et fichiers, y compris les entrées `.` et `..`. Ce numéro est en premier dans l'affichage.

Le travail consiste à comprendre que `.` et `..` sont des liens vers les dossiers qui les contiennent et le parent.

Par exemple, on voit que le dossier `.` dans `d1` a le même numéro d'i-node que `d1` et que `..` dans `d2`. Ça veut dire que ce sont les mêmes choses, mais à des endroits différents. Faites pareil avec le dossier `.` de `d2` : constatez qu'il a le même numéro d'i-node qu'un autre dossier.

Pour finir, analyser le nombre de liens de `d2`, pourquoi a-t-il 2 liens : lesquels ? pourquoi 3 liens sur `d1` : lesquels ?

## **2) Gestion des comptes utilisateurs**

Cette partie concerne la gestion des comptes pour expérimenter à la fois les commandes d'administration et les protections des fichiers.

Relisez le cours si vous avez été absent ou endormi pendant l'amphi...

### **a) Création de groupes**

Créez trois nouveaux groupes : `clients`, `visiteurs`, `programmeurs`. Laissez les GID proposés par défaut par le système, probablement 1001, 1002 et 1003 sachant que le groupe du compte `lili` est `administrateurs` et déjà créé avec le GID 1000.

Affichez le contenu du fichier `/etc/group` et retrouvez les informations concernant ces trois groupes ainsi que le groupe nommé `administrateurs` dont fait partie le compte `lili`.

### **b) Création de comptes**

Commencez par étudier les informations de votre propre compte sur votre système virtuel : tapez la commande `id`. Elle affiche que votre login est `lili` et votre UID est 1000, vous appartenez au groupe `administrateurs`, et à plusieurs autres groupes.

Normalement, la commande `egrep lili /etc/group` doit afficher tous ces groupes, sauf le groupe principal qui, lui, est inscrit dans `/etc/passwd`.

#### **i) Création de trois comptes**

Maintenant, créez trois nouveaux comptes, pour chacun le système vous demandera toutes les informations dont le mot de passe. Mettez exactement le même mot de passe pour les trois comptes (il ne faut surtout pas faire ça en vraie situation), par exemple « TP11! ».

- `bill` dans le groupe `visiteurs`, son nom en clair est Bill Gates, de Microsoft
- `jack` dans le groupe `clients`, son nom en clair est Jack Nickolson de Sunset Rise

- ada dans le groupe programmeurs, son nom en clair est Ada Lovelace de Ockham Park.

Affichez les informations de chaque compte avec la commande `id` suivie du nom du compte à afficher. Affichez le contenu des fichiers : `/etc/passwd` et `/etc/shadow` et cherchez ce qui concerne ces nouveaux comptes. Remarquez que vous avez mis le même mot de passe mais le cryptage dans `/etc/shadow` n'est pas le même ; c'est dû à la « pincée de sel » qui se trouve juste après le `$6$`, c'est un code aléatoire qui rend le cryptage des mots de passe toujours différent.

Faites `ls -l /home` et constatez que les home dirs ont été créés et sont attribués à ces utilisateurs. Le contenu initial vient de `/etc/skel`.

#### ii) Connexion sur les comptes

Tentez une connexion sur le compte de bill, il y a deux possibilités que vous allez essayer : soit `su bill`, soit `su - bill`. La différence entre les deux commandes, c'est qu'elle lance ou non les scripts d'initialisation du compte à la connexion : `.profile` et `.bashrc`, et qu'on change ou non de dossier courant. On se déconnecte par `exit` ou `logout` ou `CTRL-D` selon la façon dont vous vous êtes connecté.

Affichez qui vous êtes avec la commande `whoami`. Listez les utilisateurs connectés avec `who`, (ici, il n'y en n'a qu'un, c'est vous).

#### iii) Ajout dans le groupe des sudoers

Connectez-vous sous le compte de bill : `su - bill` puis tentez de faire `sudo cat /etc/shadow`. Vous allez recevoir un petit cours de morale et aussi un avertissement (mais c'est pour la bonne cause, n'importe qui n'est pas autorisé à tout changer dans le système).

On décide de faire confiance à Ada, vu qu'on lui doit la notion de programmation, alors on va lui permettre de taper `sudo`. C'est le fichier `/etc/sudoers` qui gère cette information, mais il est extrêmement sensible à toute erreur. Si vous en faites une, alors vous-même ne pourrez jamais plus faire `sudo`... alors veillez à ce que vous faites. Si jamais ça arrive, il vous faudra supprimer puis recréer le disque persistant de la machine virtuelle puis tout recommencer à zéro.

Comme une erreur est toujours possible et qu'elle est extrêmement grave : impossibilité définitive de faire `sudo` par la suite, je vous propose d'ouvrir une 2<sup>e</sup> fenêtre bash, et d'éditer le fichier dedans sans quitter l'éditeur : faire `CTRL-O` avec nano mais pas `CTRL-X` et `:w!` avec vi mais pas `ZZ` ou `:q`

Éditez le fichier `/etc/sudoers` et rajoutez les lignes suivantes tout à la fin :

RAPPEL : NE QUITTEZ PAS L'ÉDITEUR !!! GARDEZ-LE OUVERT DANS UNE SECONDE FENÊTRE, AFIN DE POUVOIR CORRIGER LE FICHIER SI JAMAIS LE SYSTÈME VOUS SIGNALE UNE ERREUR.

```
# ada est gentille, alors elle a le droit de tout faire
ada    ALL=(ALL) ALL
```

NB : vi passe son temps à prévenir que ce fichier est en lecture seule, forcez la sauvegarde avec `:w!`

Maintenant, connectez-vous sous le compte ada et tentez de faire `sudo cat /etc/shadow`.

Si ça marche, alors vous pouvez quitter l'éditeur, sinon remettez tout dans l'état.

#### iv) Ajout de groupes secondaires

Rajoutez ada dans le groupe visiteurs et dans le groupe clients (relisez le cours sur les groupes secondaires). Affichez à nouveau ses identifiants pour vérifier. Rajoutez jack dans le groupe visiteurs et vérifiez cet ajout.

Le fait d'appartenir à plusieurs groupes permet d'avoir des droits supplémentaires ; c'est ce qu'on va maintenant étudier.

### 3) Protection des fichiers et dossiers

La commande `ls -l` affiche les protections des fichiers et dossiers et aussi leur propriétaire et le groupe dans lequel on a mis ces éléments. NB : le groupe d'un fichier peut être différent du groupe du propriétaire.

#### a) Droits par défaut à la création d'un fichier

Ouvrez une nouvelle fenêtre terminal et connectez-vous dans le compte de ada par `su - ada`. Si vous êtes en mode console (pas de `startx`), alors déconnectez-vous d'abord du compte Lili.

Créez un fichier et un dossier vide : `touch essai1 ; mkdir dir1`.

Affichez leurs droits : `ls -l` Vous allez voir `rw-r--r--` pour le fichier et `rw-r-xr-x` pour le dossier.

Ces droits, mis par défaut, sont définis par la commande `umask`. Tapez `umask` sans paramètre : on voit `0022`. Le premier 0 indique le mode octal (base 8). La suite `002` signifie que partant de `rw-`, on n'enlève rien à la catégorie U et seulement le droit W (R=4, W=2 et X=1) aux catégories Group et Other.

Changez le `umask` pour `027` en faisant :

```
umask 027
touch essai2 ; mkdir dir2
ls -l
```

Puis mêmes choses avec `077` (`essai3`, `dir3`), `277` (`essai4`, `dir4`) et pour finir `777` (`essai5`, `dir5`). Observez que les droits sont progressivement supprimés, mais on peut les remettre à tout moment tant qu'on est le propriétaire. Remettez `umask` à `0022` puis supprimez tous les fichiers `essai` et dossiers `dir`.

#### b) Droits pour soi-même sur un fichier

Maintenant, on va définir les droits sur un dossier et un fichier et observer si différentes actions sont autorisées ou interdites. Créez un dossier appelé `dossier` et dedans un fichier appelé `fichier`. Éditez le contenu et mettez OK dedans. Revenez au dessus du dossier.

La commande pour changer les droits : ajouter, enlever ou affecter est écrite dans le cours d'amphi. Ça fera une découverte mais une perte de temps pour tous ceux qui étaient absents en cours.

Avec les droits actuels, normalement `rwx` sur le dossier et `rw-` sur le fichier pour la catégorie U, vous devez pouvoir faire ces opérations (en étant juste au dessus du dossier) :

1. lister le dossier : `ls dossier`
2. afficher le contenu du fichier : `cat dossier/fichier`
3. modifier le contenu du fichier : `date >> dossier/fichier`
4. renommer le fichier : `mv dossier/fichier dossier/fichier2` (faire le renommage inverse)
5. copier le fichier : `cp dossier/fichier dossier/fichier2`
6. supprimer le fichier supplémentaire : `rm dossier/fichier2`

Commencez par enlever le droit `w` sur le fichier, puis retentez les 6 opérations : lesquelles sont refusées ? Lesquelles ne le sont pas ?

Ensuite, enlevez le droit `r` sur le fichier et refaites les opérations en notant lesquelles sont empêchées. Remettez le droit `r` sur le fichier.

Ensuite, enlevez le droit r sur le dossier et refaites les opérations en notant lesquelles sont empêchées. Remettez le droit r sur le dossier.

Ensuite, enlevez le droit x sur le dossier et retentez les opérations. Ne faites pas attention au message `ls : dossier/fichier : Permission denied`, il signifie que comme vous ne pouvez pas descendre dans le dossier, vous ne pouvez pas afficher le nom du fichier avec la bonne couleur, mais vous avez quand même son nom, alors que quand r est enlevé, vous ne l'avez pas. Remettez le droit x sur le dossier.

Pour finir, enlevez le droit w sur le dossier et retentez les opérations.

Tirez-en une conclusion sur la signification des droits sur les commandes et leurs paramètres. Remarquez que vous n'avez agi que sur les droits de la catégorie U, c'est à dire vous, le propriétaire du fichier ou dossier.

### **c) Droits entre différents utilisateurs**

Vous allez maintenant gérer les protections entre vos trois utilisateurs. On ne va pas pouvoir tout tester, mais le principe est le suivant : en tant qu'administrateur du système (compte lili), vous créez un dossier appelé `/home/projets` et dedans vous créez un fichier appelé `readme` vous allez faire en sorte de satisfaire différentes contraintes d'accès à ce fichier.

```
sudo mkdir /home/projets
sudo chown lili:administrateurs /home/projets
vi /home/projets/readme          (mettez bonjour dedans)
```

A l'aide des commandes `chown`, `chgrp` et `chmod`, mettez en place ces contraintes :

- Ada doit pouvoir afficher et modifier `/home/projets/readme`. Probablement, elle devra être la propriétaire du fichier en question.
- Aucune personne du groupe `visiteurs` ne doit pouvoir afficher le fichier `readme`, sauf celles qui font aussi partie du groupe `clients` : en clair, `bill` ne peut pas afficher `readme` et `jack` le peut. Donc probablement, il faut que le fichier soit placé dans le groupe `clients`.
- Les gens du groupe `clients` (`jack` et `ada`) doivent pouvoir afficher `readme` s'ils connaissent son nom car personne autre que les gens du groupe `programmeurs` (`ada`) ne doit pouvoir lister le dossier (savoir quels autres fichiers il y a dedans). Ça concerne le dossier et non pas le fichier.
- On n'a pas défini ce qui concerne les modifications dans le dossier : aucun utilisateur à part ceux du groupe `programmeurs` ne doivent pouvoir créer ou supprimer des fichiers.

Rédigez vos idées dans le wiki (commandes à taper pour réaliser ces contraintes). Et indiquez aussi les commandes à faire pour vérifier les contraintes.

## **4) Autres idées de manip liées aux comptes**

Voici quelques idées qu'on ne pourra probablement pas tester, faute de temps.

Supprimer l'un des comptes mais sans enlever son `home` dir et constatez que le nom de l'utilisateur a été remplacé par son ancien UID dans la liste de `ls -l`.

Changez le GID de l'un des groupes principaux et constatez alors que le nom du groupe n'est plus correctement affiché, on ne voit que le numéro, dans la liste de `id`.

Changez l'UID de l'un des comptes en éditant à la sauvage le fichier `/etc/passwd`. Constatez alors que son `home` dir ne lui appartient plus. Quelles sont les commandes qu'il faut faire pour réparer cette situation ?

## **5) Fin du TP**

Déposez votre wiki sur l'ENT.

Fermez la machine virtuelle proprement puis quittez le logiciel VMware.