TP n°6 – Système de fichiers

1 Révisions sur les permissions

- **Q1.** Créer un répertoire `tp6` dans vos documents (et pas ailleurs !). Se placer dans ce dossier et créer un fichier `test.txt`. Écrivez quelquechose dedans (n'importe quoi).
- **Q2.** Afficher les permissions de ce fichier.

On rappelle qu'on modifie les permissions d'un fichier ou d'un dossier avec `chmod`. On indique `u`, `g`, `o`, `a` selon qu'on veut modifier les permissions de l'utilisateur, des groupes de l'utilisateur, des autres ou de tout le monde. On indique si on veut ajouter, supprimer ou fixer exactement des permissions avec +, − ou =. On indique par `r`, `w`, `x` les permissions qu'on veut modifier. On indique aussi le fichier ou le dossier dont on veut modifier les permissions.

Par exemple : `chmod u=rx test.txt` ou `chmod g-r fichier` ou `chmod u+x,g+r mon_dossier`.

- **Q3.** Donner, à tout le monde, les droits en lecture seule uniquement au fichier `test.txt`. Ouvrir le fichier, écrire quelquechose et sauvegarder. Que se passe-t-il ?
- **Q4.** Donner, à l'utilisateur uniquement, les droits en écriture, et seulement cela. Ouvrir le fichier. Que se passe-t-il ?
  Redonner les droits en lecture et écriture pour l'utilisateur.

La commande `cp` permet de créer une copie d'un fichier. On l'utilise ainsi : `cp chemin/fichier1 chemin/fichier2`, où `fichier1` est le fichier d'origine et `fichier2` est la copie. Le chemin peut permettre de mettre la copie dans un autre dossier par exemple.

La commande `rm` permet de supprimer des fichiers (attention !!). On l'utilise ainsi : `rm chemin/fichier`, où `chemin` est le chemin vers le fichier et `fichier` est le nom du fichier.

- **Q5.** Copier avec la commande `cp` le fichier `test.txt` en un fichier `test1.txt` et en un fichier `test2.txt` dans le même répertoire. Supprimer les droits en écriture du fichier `test1.txt` et essayer de le supprimer. Que se passe-t-il ? Répondre `y` (yes) puis appuyer sur entrée.
- **Q6.** Créer un sous-dossier `dossier-test`. Afficher ses permissions.
- **Q7.** A quoi peuvent correspondre les droits de lecture, d'écriture et d'exécution pour un dossier ? Pour le savoir, changez ses droits et testez si vous pouvez l'ouvrir avec `cd`, regarder son contenu avec `ls` (depuis le répertoire `tp6`, exécutez `ls dossier-test`), créer un fichier à l'intérieur, supprimer des fichiers, supprimer `test` lui-même, copier un fichier contenu dans `test` ailleurs ...
- **Q8.** Remettre les droits de `dossier-test` à la normale.

2 Liens physiques et liens symboliques

On s'intéresse ici au stockage sous forme d'inode des fichiers. On rappelle qu'on peut afficher l'inode d'un fichier avec l'option `-i` de la commande `ls`.

- **Q9.** On utilise alors la commande `ls -li` pour obtenir toutes les informations sur un fichier. Quel est le numéro d'inode du fichier `test.txt` ?
  Quel est le numéro d'inode du fichier `test2.txt`, qui, rappelons-le, est une copie du fichier test.txt.
- **Q10.** Modifier le contenu du fichier `test2.txt`. Est-ce que le fichier `test.txt` a été modifié ? (Normalement non, ils n'ont pas le même inode, ce sont donc deux fichiers indépendants)

On va expérimenter la création de la commande `ln` qui crée un lien physique vers un fichier.

- **Q11.** Dans le répertoire `tp6`, exécuter `ln test.txt test3.txt`. Quel est l'inode de `test3.txt`, à comparer avec ceux de `test.txt` et `test2.txt` ?
- **Q12.** Le chiffre qui suit les droits et précède le nom du propriétaire du fichier affiché par la commande `ls` indique le nombre de liens physiques (c'est aussi le nombre de noms différents) d'un fichier. Quel est ce chiffre pour le fichier `test.txt` ?
- **Q13.** Supprimer le fichier `test3.txt` (avec `rm`). Est-ce que le fichier `test.txt` a été supprimé ? Quel est maintenant le nombre de liens physiques de `test.txt` ?
- **Q14.** Comment le système de fichiers peut-il savoir quand il peut réellement supprimer un fichier ?

On va maintenant créer des liens symboliques. On peut créer un lien avec l'option `-s` de `ln` et on utilise toujours `rm` pour la suppression.

- **Q15.** Lister le contenu de votre répertoire utilisateur ~ et repérer les liens symboliques.

- **Q16.** Dans votre dossier personnel, créer un lien symbolique vers le répertoire `dossier-test` qui se trouve dans le répertoire `tp6`. Vérifier que cela fonctionne correctement puis supprimer ce lien.
- **Q17.** Toujours dans votre répertoire personnel, créer un lien symbolique vers le fichier `test2.txt` que vous avez créé plus haut. Ouvrir le lien avec Vscode. Que se passe-t-il ?
- **Q18.** Supprimer le fichier `test2.txt` qui se trouve dans `tp6`. Que se passe-t-il pour le lien symbolique ? Ouvrez-le avec VScode, écrivez quelquechose et essayez d'enregistrer. Si vous retournez dans `tp6`, `test2.txt` devrait être revenu.
- **Q19.** Supprimez le lien.

3 Redirections

Pour illustrer cette partie on va utiliser quelques nouvelles commandes. (pas besoin de se souvenir d'elles au-delà de ce TP)

- `cat` qui permet d'afficher le contenu d'un ou plusieurs fichiers dans le terminal.
- `wc` qui peut compter le nombre de lignes, de caractères ou d'octets d'un fichier.
- `less` peut afficher un très très très long texte dans le terminal en permettant à l'utilisateur de se déplacer dans le texte. Le terminal a une quantité de place limitée, donc un très très très long fichier ne peut pas être affiché, le début est tronqué.

- **Q20.** Tester les commandes `cat test.txt`, `wc test.txt` et `less test.txt` (appuyer sur q pour revenir à l'état normal).
- **Q21.** Essayer d'afficher le texte intégral du livre 1 des Misérables avec `cat`. Scroller aussi haut que possible. Êtes vous au début du livre ?
- **Q22.** Pour résoudre ce problème, on utilise `less`. Redirigez la sortie de `cat` dans l'entrée de `less` avec un tuyau. Maintenant vous pouvez lire le début et la fin du livre.
- **Q23.** Écrire un petit code C qui récupère un entier n via scanf et affiche "il y a n lignes".
- **Q24.** En utilisant une redirection < et un fichier, faire afficher au programme "il y a 120 lignes", **sans taper le 120 au clavier**.
- **Q25.** Écrire une commande bash qui affiche "il y a n lignes" avec n le nombre de lignes du livre 1 des Misérables. (indice : utiliser `wc` avec la bonne option, le programme précédent et un tuyau)

4 Montage d'un support de fichier

Le système de fichiers est composé de plusieurs sous-systèmes de fichiers assemblés les uns avec les autres. Certains sous-systèmes de fichiers peuvent se trouver sur le disque dur de la machine, d'autres sur un serveur et d'autres peuvent être amovibles (par exemple une clé USB).

Ces sous-systèmes, qui suivent eux-mêmes une structure arborescente, sont rattachés à l'arborescence générale, donnant ainsi l'illusion d'une unique arborescence.

La commande `mount` permet de visualiser les systèmes de fichiers qui sont montés à un instant donné. Elle liste les noms des disques/partitions de disques/ autres puis "on" puis le point où ils sont montés dans l'arborescence, c'est à dire quel dossier est en fait sur ces périphériques. Le reste de l'affichage est des options qu'on ignorera.

- **Q26.** Exécuter la commande `mount` sans arguments et observer le résultat.
  Où est montée la partition de disque `/dev/mapper/debian-vg-root` ? Qu'est-ce que cela veut dire ?
  On voit que le disque dur `/dev/sda1` (1ère partition du premier disque dur) est monté en `/boot`, c'est à dire contient les instructions de mise en route de Linux.
  On voit également que le système de fichier `/nfs` du serveur `10.121.13.1` est monté sur le répertoire `/nfs` de la machine. Les documents stockés sur votre session ne sont pas stockés sur l'ordinateur mais sur ce serveur, et le système de fichiers qu'on utilise est en fait un système en réseau (c'est ce que signifie nfs).
- **Q27.** Insérer une clé USB, double-cliquer sur l'icone apparaissant sur le bureau et observer avec `mount` les modifications que cela induit.

Lorsqu'une clé USB est insérée, il faut qu'elle soit "montée" pour pouvoir accéder à son contenu. On définit un point dans l'arborescence des dossiers et fichier auquel on pourra trouver les fichiers de la clé.

- **Q28.** Noter le point de montage de votre clé (info à droite du "on") et quel est son nom de partition d'après Linux (info à gauche du "on", probablement `/dev/sdb1` mais cela peut changer selon l'ordinateur)
- **Q29.** Démonter la clé USB : `umount (point de montage de la clé)`. Vérifier que cela a marché avec la commande `mount`.

On peut également utiliser la commande `mount` pour gérer manuellement le montage d'un système de fichiers :

- **Q30.** Cette fois on va monter à la main la clé sur le système de fichiers. Créer un dossier `Cle` dans vos documents pour servir de point de montage (le point précédent ayant été supprimé).
- **Q31.** Effectuer la commande `mount -t vfat (nom de partition de la clé) (chemin vers le dossier Cle)`. Une erreur en français apparait, que s'est il passé ?
- **Q32.** Si vous êtes sur votre propre ordinateur, corrigez l'erreur.

À défaut de pouvoir utiliser `mount` sur les ordis du lycée, on peut utiliser la commande `udisksctl`, qui ne nous permet pas de choisir où la clé est montée.

- **Q33.** Utilisez la commande `udisksctl mount -b (nom de la clé d'après Linux)` et vérifiez que vous avez maintenant accès au contenu de la clé. Où est-elle montée ? Qu'est-ce que cela nous indique sur ce que le double clic sur l'icone de la clé fait en réalité ?