TP 2

Notions et commandes abordées
- commandes : chmod, cat, more, less, grep, cut, find, tail ;
- les redirections et le pipe.

Trucs et astuces
- Documentation des commandes : la commande `man` permet d'accéder à la documentation des commandes installées sur le système. Si vous rencontrez un problème de syntaxe pour une commande, ou que vous avez oublié l'option magique d'une commande, n'hésitez à taper `man <commande>`.

Chmod, le programme pour jouer avec les permissions

1. Ouvrez un terminal, allez dans votre répertoire Systeme.
2. Créez un répertoire `tp2` et allez dedans.
3. Créez 3 fichiers `test1`, `test2.txt` et `yop.txt`.
4. En faisant `ls -al`, vous devriez avoir :

```
  total 0
drwxr-xr-x  2 beyler prof 128 2005-10-07 15:45 .
drwxr-xr-x  3 beyler prof  72 2005-10-07 15:45 ..
-rw-r--r--  1 beyler prof   0 2005-10-07 15:45 test1
-rw-r--r--  1 beyler prof   0 2005-10-07 15:45 test2.txt
-rw-r--r--  1 beyler prof   0 2005-10-07 15:45 yop.txt
```

   La première colonne représente les droits de lecture, d'écriture et d'exécution d'un fichier. On distingue 10 caractères pour chaque fichier.
   - le premier caractère représente le type de fichier. Un `-` représente un fichier normal, un `d` représente un repertoire (ou dossier). Il en existe quelques autres que nous n'aborderons pas ici ;
   - les 3 caractères suivants sont les droits du propriétaire du fichier. Est-ce que le propriétaire peut lire, modifier ou exécuter un fichier ?
   - ensuite, nous avons les permissions pour les utilisateurs du groupe du propriétaire ;
   - enfin, nous avons les droits pour tous les autres utilisateurs ;
   - les autres colonnes sont expliquées dans la page manuel de la commande `ls`.
5. En regardant la page manuel (utilisant la commande `man`) de la commande `chmod`, changez les droits d'accès du fichier `test1.txt` pour que personne sauf vous ne puisse le lire.
6. Changez maintenant les permissions du fichier `yop.txt` pour que le groupe soit le seul à pouvoir le lire et le modifier.
7. Enlevez le droit d'écriture au groupe pour le fichier `yop.txt`
8. Vérifiez si les changements ont été pris en compte.
9. Remettez ensuite les droits par défaut (lecture/écriture pour vous et lecture pour le groupe et les autres).
10. Il existe aussi une autre solution pour définir les droits d'accès. À la place d'utiliser une commande de type `chmod ugo=rwx yop.txt`, nous pouvons utiliser une version numérique. En effet, si on suppose que les permissions d'un fichier sont divisées en trois catégories (propriétaire, groupe et autre) et que la lecture, l'écriture et l'exécution ont respectivement les valeurs 4, 2 et 1 alors on peut faire la somme des droits par catégorie ce qui donne un chiffre de 0 à 7.

Exemple :

`chmod ugo=rwx yop.txt` devient `chmod 777 yop.txt`.

Refaites les modifications de droits utilisant la version numérique.

Remarque : la commande `chmod og+=rw yop.txt` ne peut pas être mise en version numérique puisque nous ne nous préoccupons pas des droits précédents.

11. Pourquoi est-ce qu'une somme marche ? N'y-a-t-il pas de risque qu'on obtienne la même somme pour des droits différents ?

Il existe des droits sur les dossiers. Les droits sur un dossier sont importants car ils ont une conséquence sur la gestion des fichiers et dossiers qu'il contient.

1. Éditez le fichier `test1` et mettez-y "L'informatique c'est facile !". Mettez-vous dans le répertoire parent (donc dans le répertoire Systeme).
2. Enlevez tous les droits sur le dossier `tp2` et essayez de faire un `ls tp2`, d'éditer avec `vi tp2/test1` et d'y ajouter une ligne et d'effacer le fichier `test2.txt` avec `rm tp2/test2.txt`.
3. Il existe 8 (`aucun,x,w,wx,r,rx,rw,rwx`) droits différents qu'un dossier/fichier peut avoir, en parcourant toutes les possibilités des droits sur le dossier regardez l'effet que cela a sur les trois manipulations précédentes. Si vous avez réussi à effacer le fichier `tp2/test2.txt`, n'oubliez pas de le recréer avant de passer au prochain test !
4. Quelles sont donc les significations des droits d'écriture, de lecture et d'exécution sur un dossier vis-à-vis des fichiers et dossiers qu'il contient ?

Les programmes `cat, more, less, find, grep` et les redirections

Les programmes qui tournent sur un système d'exploitation ont toujours une *entrée standard, sortie standard et la sortie erreur*. Lorsque vous avez un programme comme `ls` qui vous affiche le contenu d'un répertoire, il utilise la sortie standard. Il est possible de rediriger ces trois éléments pour obliger le programme à écrire dans un fichier ou lire à partir d'un fichier.

<table>
<tr><td>;</td><td>séparateur de séquence inconditionnel</td></tr>
<tr><td>&&</td><td>et logique</td></tr>
<tr><td>||</td><td>ou logique</td></tr>
<tr><td><</td><td>redirection de l'entrée standard (stdin)</td></tr>
<tr><td>></td><td>redirection de la sortie standard (stdout)</td></tr>
<tr><td>2></td><td>redirection de l'erreur (stderr)</td></tr>
<tr><td>»</td><td>redirection de stdout en fin du fichier</td></tr>
<tr><td>|</td><td>enchainement de commande (stdout vers stdin)</td></tr>
</table>

1. Essayez `man ls > tmp`.
2. Le programme `cat` affiche le contenu d'un fichier. Tapez `cat tmp`. Que contient le fichier `tmp` qui vient d'être créé ?
3. Les programmes `more` et `less` sont deux autres afficheurs. Testez les.
4. Le programme `find` permet de trouver un fichier dans une arborescence.

   Tapez `find /usr/NX -name "lib*"`. Que fait cette commande ?

   Vous devriez voir des lignes qui ressemblent à *find : /usr/NX/var/db/running : Permission denied*, c'est en fait une message d'erreur. À l'aide du tableau ci-dessus, redirigez la sortie d'erreur vers le fichier `/dev/null` qui est un fichier poubelle.
5. Revenons sur les options de la commande `ls`. Supposons que nous savons qu'une option qui affiche tous les fichiers existe mais nous avons oublié comment elle s'appelle. Une solution est de chercher dans la page `man` pour la trouver. Une autre est d'utiliser le programme `man` couplé avec le programme `grep` qui permet d'afficher les lignes qui contiennent une partie d'une chaîne de caractères dans un fichier.

   Exemple :

`grep "bonjour" test.txt` affichera les lignes du fichier `test.txt` qui contiennent la chaîne de caractères "bonjour".

Utiliser donc la commande `grep` sur le fichier `tmp` pour trouver l'option à utiliser qui nous donnera un affichage de tous les fichiers (on pourra chercher dans le fichier la chaîne "tous" par exemple).

Remarque : Une option intéressante de grep est de donner les lettres ou chiffres qui nous intéressent. Ainsi, `grep [a-d] tmp` afficherait les lignes du fichier `tmp` qui contiennent les lettres a jusqu'à d. Autre exemple, `grep [0-9] tmp` affiche les lignes contenant des chiffres.

6. Une dernière solution pour trouver ces options est d'utiliser un *pipe*. En fait, on redirige la sortie standard d'un premier programme pour qu'il soit directement l'entrée standard d'un deuxième programme.

   Tentez la commande `man ls|grep "tous"`.

Les programmes `cal, tail, cut et grep`

Cet exercice permet de montrer l'utilisation avancée du pipe. N'oubliez pas d'utiliser le pages manuels pour les programmes que vous ne connaissez pas.

1. Exécutez le programme `cal` pour afficher le mois de juin 2006.
2. En utilisant les programmes `tail` et `cut`, on peut enlever les deux premières lignes et extraire les colonnes du mercredi du mois.
3. Si vous regardez bien, il y a des lignes ne contenant aucun nombre, utiliser `grep` pour enlever ces lignes.
4. En utilisant en dernier le programme `wc`, affichez le nombre de mercredis du mois de juin 2006.