T.D. 3 - Le Shell (première partie)

1 Manipulations avec l'éditeur de textes

Vous êtes connus du système par un *nom de login* cryptique. Le fichier `/etc/passwd` contient des informations selon le format ci-après :

nom de login:*password*:*uid*:*gid*:*nom en clair*:*home dir*:*shell*

- *password* est le mot de passe crypté (de plus en plus de systèmes n'ont qu'une étoile dans ce champ),
- *uid* et *gid* sont les identificateurs numériques de l'utilisateur et du groupe auquel il appartient,
- *nom en clair* est une zone de commentaire dans laquelle les administrateurs système mettent votre vrai nom,
- *home dir* est votre répertoire par défaut, correspondant à la variable `HOME`,
- *shell* est le chemin de votre *shell* par défaut.

Copiez le fichier `/etc/passwd` dans votre répertoire par défaut. Vérifiez avec la commande `ls -l` que vous pouvez modifier la copie que vous en avez faite. Changez éventuellement les droits d'accès au fichier avec la commande `chmod`.

À l'aide de l'éditeur de textes `vi`, donnez la suite minimum de commandes de substitution pour ne conserver dans ce fichier que les *noms en clair* des utilisateurs.

2 La commande `grep`

Les utilisateurs appartiennent à un groupe. Celui-ci est identifié par un nombre (le *gid* pour *group identifier*) dans le fichier `/etc/passwd`.

À l'aide de la commande `grep`, obtenez la liste des lignes dans `/etc/passwd` qui correspondent aux utilisateurs d'un groupe donné par son numéro. On prendra soin à ne pas afficher les lignes contenant ce numéro en position autre que *gid*.

Comment faire pour obtenir cette liste dans un fichier placé dans votre répertoire courant ?

3 Tubes

Quel est le résultat de la commande ci-dessous ? Expliquez en détail son fonctionnement.

```
ls -l /usr/lib | grep '^d'
```

Comment obtenir l'effet inverse ?

4 Les variables d'environnement

Ouvrez une nouvelle fenêtre *shell*. Dans cette nouvelle fenêtre, affichez le contenu de la variable `HOME`. Changez la en `/usr/lib`, puis faites `cd` sans argument. Que se passe-t-il ?

Faites à présent `cd` sans argument dans une autre fenêtre (celle d'origine par exemple). Quelle conclusion en tirer ?

Les commandes se trouvent dans des fichiers exécutables. Ces fichiers sont dans des répertoires dont le nom figure dans votre variable `PATH`. Comment faire pour chercher où se trouve la commande `ls` en ne modifiant que la variable `PATH` (faites cet exercice dans la nouvelle fenêtre) ?

5 La commande `wc`

À l'aide de la commande `wc`, comptez le nombre de lignes dans le fichier `/usr/include/stdio.h`.

Toujours à l'aide de cette commande, comptez le nombre de fichiers ou répertoires dans le répertoire courant. Cette dernière fonctionnalité étant tellement pratique (si, si...), vous allez placer votre solution dans un fichier que vous allez rendre exécutable. Exécutez-le. Félicitations, vous avez créé un *script shell* !

6 La commande `find`

La commande `find` permet de faire des recherches dans l'arborescence du système de fichiers.

La forme générale de la commande est :

> `find` *répertoire prédicat prédicat ...*

Le *répertoire* est le répertoire à partir duquel commence la recherche. Les *prédicats* sont des sélecteurs : pour qu'un fichier soit sélectionné, il faut que tous les prédicats appliqués à ce fichier soient vrais. Par exemple `find . -name toto -print` recherche, à partir du répertoire courant (le "`.`" signifie *répertoire courant*), tous les fichiers de nom `toto`. Le prédicat `-print` est spécial : il est toujours vrai, mais son rôle est d'afficher le fichier sélectionné.

Comment faire pour afficher tous les fichiers et répertoires qui se trouvent *sous* votre répertoire initial (à quelque niveau que ce soit) ? Faites de cette commande un script.

À l'aide du manuel de la commande `find`, trouver une manière d'afficher tous les répertoires qui se trouvent sous votre répertoire courant.

Comment obtenir le nombre total de fichiers et répertoires dans toute l'arborescence du système ? Exécutez cette commande. Quelle est votre conclusion ?