Exercice : Commandes de base linux

Vous trouverez de l'aide sur les commande linux sur le site : Les commandes de base en console linux

Exercice 1

On suppose que le répertoire personnel de l'utilisateur courant est vide.

1) Décrire l'effet des commandes suivantes en supposant qu'elles sont exécutées les unes après les autres dans cet ordre.

- **a)** cd
- **b)** mkdir NSI
- **c)** mkdir NSI/TP_SHELL
- **d)** cd NSI/TP_SHELL
- **e)** ls
- **f)** cd ..
- **g)** mkdir PYTHON
- **h)** ls -l
- **i)** chmod u+rwx,g-rwx,o-rwx TP_SHELL

2) Représenter l'arborescence de fichiers du dossier utilisateur.

Exercice 2

On utilisera les commandes suivantes :

— La commande touch nom_fichier permet de créer un fichier vide en donnant un nom au fichier.
— Le programme nano nom_fichier est un éditeur de texte qui permet de visualiser et modifier un fichier.
— La commande rm nom_fichier permet de supprimer le fichier indiqué après la commande.
— La commande echo message affiche le message indiqué après la commande.

1) Placez vous dans le répertoire TP_SHELL et créer un fichier vide nommé exolinux.txt.

2) Editez le fichier exolinux.txt et insérer le texte suivant : "Je peux tout faire avec les commandes linux." Sauvegarder et quitter l'éditeur.

3) Afficher votre fichier dans la console avec la commande cat.

4) Exécuter la commande echo "bonjour". Que se passe-t-il ?

5) Exécuter la commande echo bonjour > exolinux.txt. Que se passe-t-il ?
Afficher votre fichier exolinux.txt. Que remarquez-vous ?

6) Exécuter la commande echo au revoir >> exolinux.txt. Que se passe-t-il ?
Aller voir votre fichier exolinux.txt.

7) Déplacer le fichier exolinux.txt dans le dossier Documents.

8) Supprimer le fichier exolinux.txt.

Exercice 3

Cet exercice devra être réalisé sur Windows et sur Linux.

Python dispose d'un module qui permet d'exécuter des commandes au niveau du système d'exploitation. Ce module est os.

La méthode system de ce module permet d'exécuter une commande.
Par exemple :

— sur windows, **os.system("C :/chemin/vers/programme")** ouvre le bloc notes ;
— sur Linux, os.system("/chemin/vers/programme") ouvre l'éditeur de texte.

On récupère le prompt de l'interpréteur python seulement en fermant l'application .

La méthode startfile permet l'ouverture d'un fichier.
Elle prend en argument le chemin complet du fichier à ouvrir.

D'autres méthodes utiles sont données ci-après (attention aux arguments) :

— name renvoie le nom de l'OS.
— getcwd() renvoie le répertoire courant.
— listdir() renvoie le contenu du répertoire courant.
— mkdir crée un répertoire.
— chdir change de répertoire courant
— remove supprime un fichier.
— rmdir supprime un répertoire.

Ouvrir l'interpréteur python et importer le module os. Toutes les actions demandées ci-après se font exclusivement en python.

1) Afficher le nom du système d'exploitation.
2) Quel est le répertoire courant ?
3) Lister le contenu du répertoire courant.
  a) Combien y a-t-il de dossiers et fichiers dans le répertoire courant ?
  b) Un fichier et un dossier sont cachés si le nom commence par un point.
  Pouvez-vous trouver combien il y en a (en python bien sur) ?
4) Créer un répertoire titi dans le répertoire courant puis placez-vous dans le répertoire titi.
5) Créer un fichier texte test.txt contenant un message de bienvenue.