

Exercice 3 (3 points)

Cet exercice porte sur les systèmes d'exploitation, la programmation en général et les dictionnaires.

Dans un système d'exploitation de type UNIX, on considère l'arborescence suivante dans laquelle les noms des répertoires sont en gras et ceux des fichiers sont en italique :

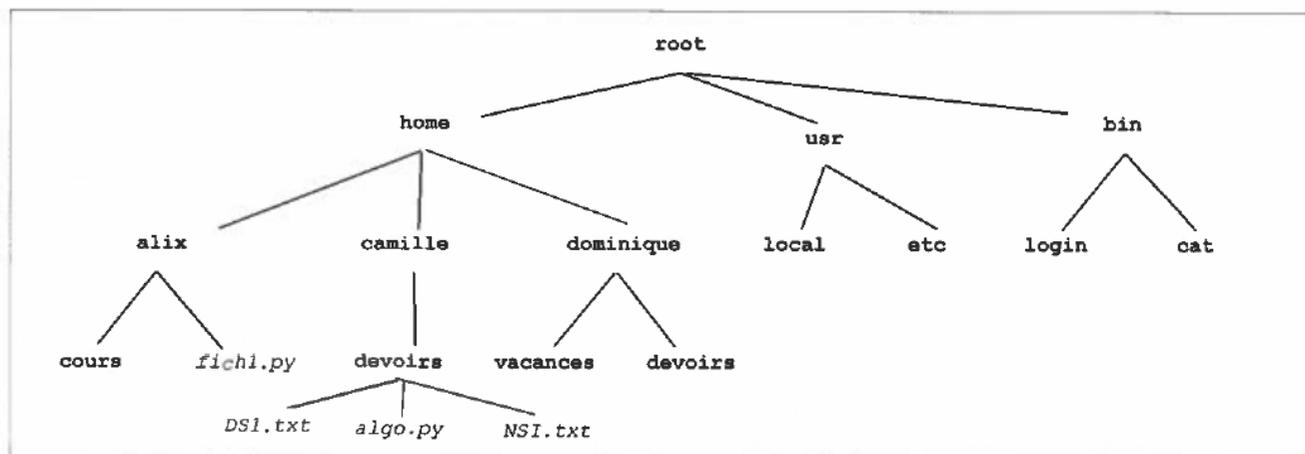


Figure 1

On donne ci-après un extrait de quelques commandes de base à utiliser dans un terminal sous linux :

Extrait des commandes de base linux

```
ls : affiche le contenu d'un répertoire.
    ls repertoire1 affiche le contenu de repertoire1
cd : permet de changer de répertoire courant.
    Après la commande cd repertoire1, le répertoire courant devient
    repertoire1
cp : crée une copie d'un fichier.
    cp fichier1.py fichier2.py copie fichier1.py sous le nom
    fichier2.py
mv : déplace ou renomme un fichier ou un répertoire.
    mv fichier.txt dossier1 déplace fichier.txt dans dossier1
rm : supprime un fichier ou un répertoire.
    rm mon_fichier.mp3 supprime mon_fichier.mp3
mkdir : crée un répertoire.
    mkdir dossier2 crée le répertoire dossier2
cat : affiche le contenu d'un fichier.
    cat fichier1.txt affiche le contenu de fichier1.txt
touch : crée un fichier qui n'existait pas jusqu'alors.
    touch fichier2 crée fichier2
```

Dans l'arborescence décrite dans la *figure 1* ci-dessus, *alix*, *camille* et *dominique* sont respectivement les répertoires personnels des utilisateurs *Alix*, *Camille* et *Dominique*.

On souhaite explorer et modifier les répertoires et fichiers grâce à l'utilisation d'un terminal.

1. Dans cette question, la problématique des droits d'accès n'est pas prise en compte.

a) Dans l'arborescence précédente, on suppose que le répertoire courant est *Camille*.

Parmi les quatre propositions suivantes, déterminer la commande qui permet d'afficher son contenu :

Proposition 1 : `mv`

Proposition 2 : `cp`

Proposition 3 : `ls`

Proposition 4 : `rm`

b) Dans l'arborescence précédente, on suppose que le répertoire courant est *Camille*.

Parmi les quatre propositions suivantes, déterminer toutes celles qui permettent que le répertoire `cours` d'*Alix* devienne le répertoire courant :

Proposition 1 : `cd ../alix/cours` **Proposition 2 :** `cd cours`

Proposition 3 : `cd alix/cours` **Proposition 4 :** `cd /home/alix/cours`

c) Dans l'arborescence précédente, on suppose que le répertoire courant est *alix*. Déterminer l'instruction permettant de copier le fichier `fich2.txt` qui se trouve dans le répertoire d'*Alix* dans le répertoire `devoirs` de *Camille*.

2. Dans cette question, la problématique des droits d'accès est prise en compte.

Camille, depuis le répertoire `camille/devoirs` utilise la commande `ls -l` et obtient le résultat suivant :

```
-rw-r--r-- 1 root    root  1096 juil. 22 17:10 DS1.txt
-rw-r--r-- 1 camille eleve  128 juil. 22 16:03 algo.py
-r--r--r-- 1 camille eleve  128 juil. 22 18:21 NSI.txt
```

- a) Expliquer pourquoi l'instruction `rm DS1.txt` est refusée si l'utilisateur est Camille.

On donne ci-après un extrait du mode d'emploi de la commande `chmod` à utiliser dans un terminal sous linux :

La commande `chmod` de linux

`chmod` : modifie les permissions d'un fichier ou d'un dossier.

`chmod` peut prendre en paramètres les caractères suivants et un nom de fichier

- u pour l'utilisateur (user)
- g pour le groupe (group)
- o pour les autres utilisateurs (other)
- + pour ajouter des droits
- pour supprimer des droits
- r pour modifier les droits de lecture (read)
- w pour modifier les droits d'écriture (write)
- x pour modifier les droits d'exécution

Exemple :

L'instruction `chmod u+rwx o-x exo.py` ajoute sur `exo.py` des droits en lecture, écriture et exécution à l'utilisateur et supprime les droits d'exécution aux autres utilisateurs.

b) Camille souhaite interdire la lecture du contenu du fichier `algo.py` aux autres utilisateurs (même ceux de son groupe). Elle souhaite également se donner la possibilité de l'exécuter. Écrire l'instruction permettant de modifier les droits sur `algo.py`. On suppose que Camille bénéficie déjà des droits de super utilisateur.

3. Dans son répertoire `vacances`, Dominique a conservé tous les fichiers durant une année.

Dominique décide d'indexer ces fichiers selon leur type : "photo", "vidéo" ou "son", le mois de leur création (un nombre entre 1 et 12), et leur taille en kilo-octets. Puis Dominique les stocke dans une liste, implémentée par le type `list` de Python.

Un extrait de la liste est donnée ci-dessous :

```
fichiers_vacances = [  
{"nom_fichier": "plage.jpg", "type": "photo", "mois" : 7,  
"taille" : 152},  
{"nom_fichier" : "mouette.mp3", "type": "son", "mois" : 7,  
"taille" : 54},  
{"nom_fichier" : "famille1.png", "type": "photo", "mois" : 12,  
"taille" : 257},  
...]
```

a) Recopier et compléter sur la copie le code de la fonction `classement_type` qui prend en paramètre une liste `L` au format précédent et qui renvoie trois listes `photos`, `sons` et `videos`, contenant les éléments du type correspondant.

```
1 def classement_type(L):
2     photos = []
3     sons = []
4     videos = []
5     for fichier in ...:
6         if ... == "photo":
7             photos.append(fichier)
8         elif ... == "son":
9             ...
10        else:
11            ...
12    return photos, sons, videos
```

b) Dominique souhaite maintenant compter le nombre de photos datant du mois de juillet. Écrire une fonction `photo_juillet` qui prend en paramètre une liste de fichiers et qui renvoie le nombre de fichiers de type "photo" du mois de juillet.

c) À partir des fichiers de la liste `photos_vacances`, Dominique souhaite stocker le plus de fichiers possible sur une clé USB d'une capacité de 64 Go.

Proposer une démarche algorithmique pour y parvenir (on ne demande pas de la programmer en Python).