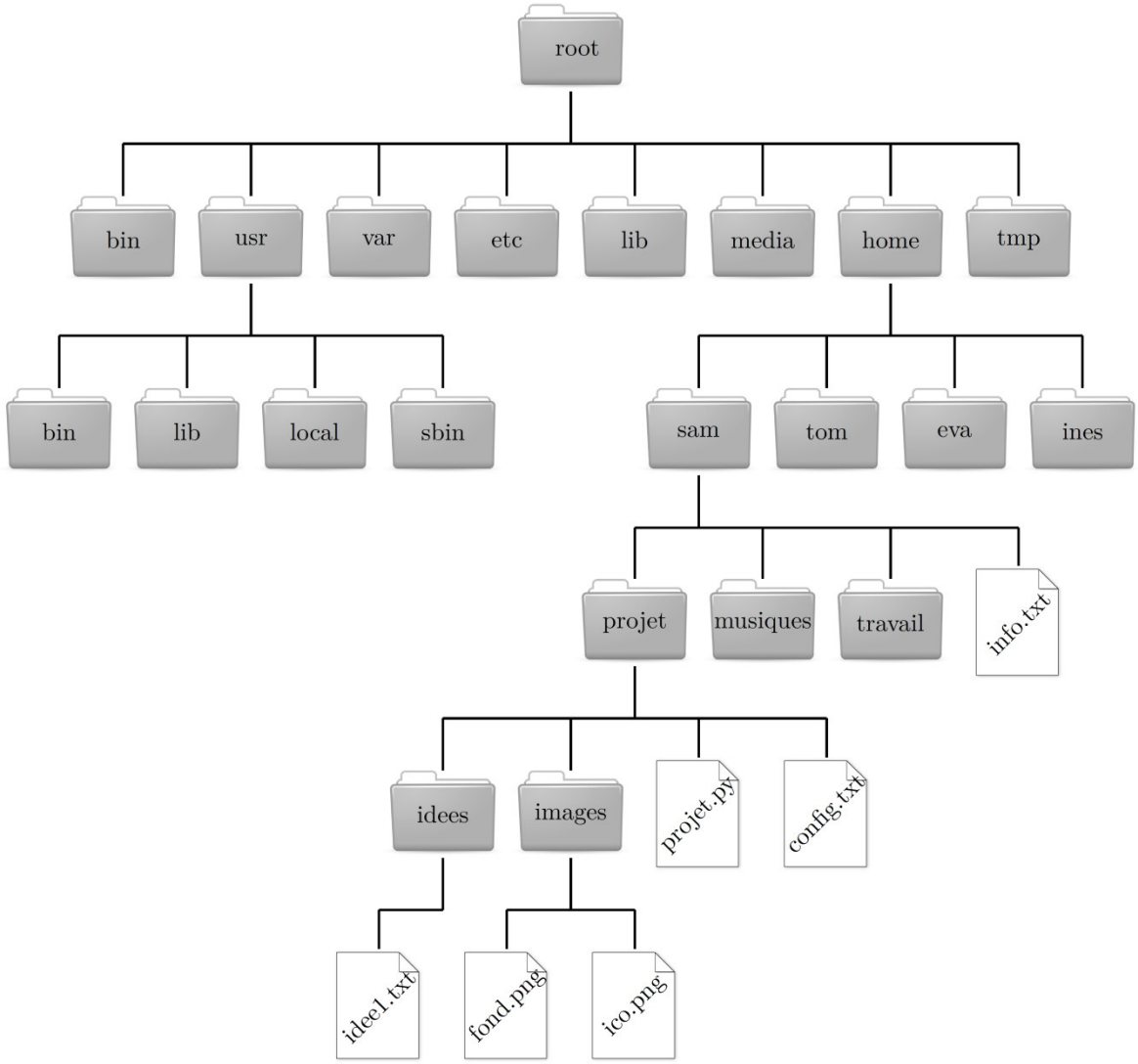
Exercice 3

Thème abordé : système d'exploitation

Nous avons l'arborescence ci-dessous sous un environnement Linux.



Samuel a pour nom d'utilisateur `sam`. Il a ouvert un terminal et le répertoire courant est le répertoire `musiques`. Pour tout l'exercice, on pourra tirer parti de l'annexe 2 répertoriant différentes commandes du système d'exploitation.

1. Ecrivez la ou les commande(s) qui permet(tent) de se déplacer du répertoire actuel `musiques` au répertoire `projet` :
   a. en utilisant un chemin relatif.
   b. en utilisant un chemin absolu.

2. Le répertoire courant est à présent le répertoire `sam`
   a. Ecrire la commande qui permet de lister le contenu du répertoire `projet`.
   b. Le fichier `config.txt` est protégé en écriture pour tous les utilisateurs. On souhaite modifier ce droit afin que l'utilisateur `sam` et lui seul puisse

modifier le contenu du fichier. Ecrire la commande permettant d'effectuer ce changement.

3. Le répertoire courant est toujours `sam`. L'utilisateur souhaite supprimer le répertoire `projet` en tapant l'instruction :

```
rm projet
```

Il constate que cette instruction ne fonctionne pas car ce répertoire n'est pas vide. *Finalement, il tape l'instruction :*

```
rm –R projet
```

où « R » signifie « récursif ». Le répertoire est finalement supprimé.

a. Pourquoi cette instruction fonctionne-t-elle, contrairement à la précédente ?

Les fichiers et dossiers ont été effacés dans cet ordre :

- fichier `idee1.txt`
- dossier `idees`
- fichier `fond.png`
- fichier `ico.png`
- dossier `images`
- fichier `projet.py`
- fichier `config.txt`
- dossier `projet`

b. Quel type de parcours a été réalisé par le système d'exploitation ?

4. On considère la fonction récursive suivante en langage Python :

```
def nb_fichiers(list_fich, i) :
  if i == len(list_fich) :
    return 0
  elif list_fich [i][0] == 'b' :
    return 1 + nb_fichiers(list_fich, i+1)
  else :
    return nb_fichiers(list_fich, i+1)
```

où `list_fich` est une liste contenant des noms de fichiers.

Indiquer ce que renvoie l'appel suivant en expliquant les étapes :

```
nb_fichiers(['nsi.bmp','banana.mp3','job.txt','BoyerMoore.py'],0)
```

Annexe 2 (exercice 3)

(à ne pas rendre avec la copie)

Extrait des commandes de base linux

`ls` *permet d'afficher le contenu d'un répertoire*
`cd` *se déplacer dans l'arborescence (ex `cd repertoire1`)*
`cp` *créer une copie d'un fichier (ex `cp fichier1.py fichier2.py`)*
`mv` *déplacer ou renommer un fichier ou un répertoire (`mv fichier.txt doss`)*
`rm` *effacer un fichier ou un répertoire (`ex rm mon_fichier.mp3`)*
`mkdir` *créer un répertoire (ex `mkdir nouveau`)*
`cat` *visualiser le contenu d'un fichier*
`chmod` *modifier les permissions d'un fichier ou d'un dossier. Pour un fichier, le format général de l'instruction est :*

```
chmod droits_user droits_group droits_other nom_fichier
```

Où `droits_user`, `droits_group` et `droits_other` indiquent respectivement les droits de l'utilisateur, du groupe et des autres et peuvent être :

```
+ ajouter
- supprimer
r read
w write
x execute
```

Exemple : `chmod rwx +r –x script.sh`