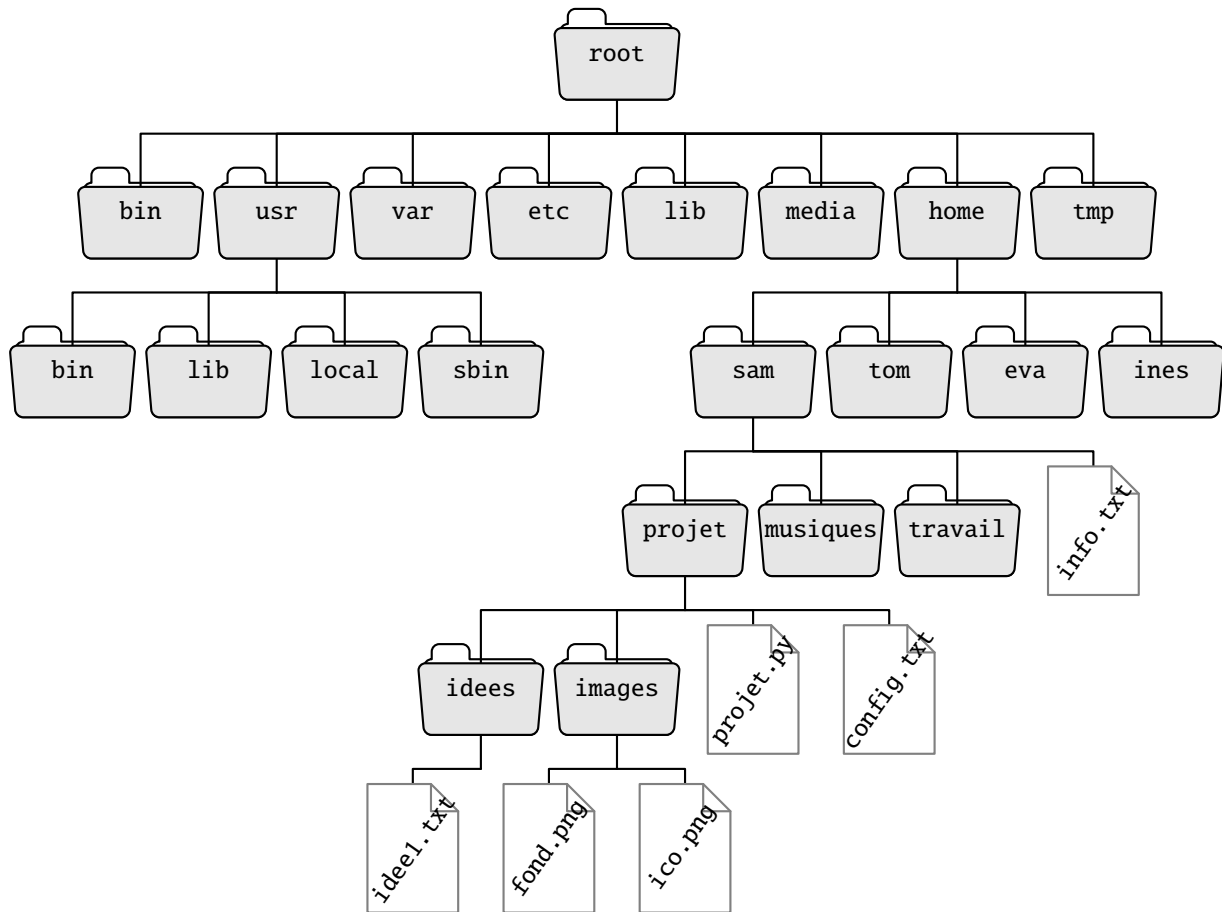


Exercices sur les OS

EXERCICE 1 : 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.

- 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) Écrire 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. Écrire 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 "récuratif". Le répertoire est finalement supprimé.

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

EXERCICE 2 : L'entreprise capNSI gère les contrats de ses clients en créant pour chacun d'eux un sous-dossier dans le dossier Contrats sur leur ordinateur central. Le système d'exploitation de cet ordinateur est une distribution linux.

Dans la console représentée sur la figure ci-dessous, on peut visualiser les répertoires (ou dossiers) à la racine de l'ordinateur central avec l'instruction ls :

```
gestion@capNSI-ordinateur_central:~$ ls
Bureau          Documents  Modèles  Public
Téléchargements  Contrats  Images  Musique
Vidéos
```

- 1) a) Donner le nom de l'utilisateur et le nom de l'ordinateur correspondant à la capture d'écran précédente.
- b) Écrire les instructions permettant d'afficher la liste des dossiers clients du répertoire Contrats en partant de la situation ci-dessous :

```
gestion@capNSI-ordinateur_central:~$
```

- 2) Après une campagne de démarchage, l'entreprise a gagné un nouveau client, Monsieur Alan Turing. Elle souhaite lui créer un sous-dossier nommé TURING_Alan dans le dossier Contrats. De plus, elle souhaite attribuer tous les droits à l'utilisateur et au groupe et seulement la permission en lecture pour tous les autres utilisateurs. La commande chmod permet de le faire.
 - a) Écrire les instructions permettant de créer le sous-dossier TURING_Alan à partir du répertoire racine.
 - b) Écrire l'instruction permettant d'attribuer les bons droits au sous-dossier TURING_Alan.

En Python, le module os permet d'interagir avec le système d'exploitation. Il permet de gérer l'arborescence des fichiers, des dossiers, de fournir des informations sur le système d'exploitation. Par exemple, le code de la page suivante, exécuté dans la console, permet de créer le sous-dossier TURING_Alan précédent :

```
>>> import os
>>> os.mkdir("Contrats/TURING_Alan")
>>> os.chmod("Contrats/TURING_Alan", 774)
```

L'entreprise dispose d'un nouveau tableau de nouveaux clients :

```
tab_clients = [('LOVELACE', 'Ada'), ('BOOLE', 'George'), ('VONNEUMANN', 'John'),
               ('SHANNON', 'Claude'), ('KNUTH', 'Donald')]
```

Elle souhaite automatiser le formatage des tableaux des nouveaux clients. Elle souhaite également automatiser la création et l'attribution des droits des dossiers portant les noms des nouveaux clients.

- 3) Écrire une fonction `formatage(tab)` qui prend en paramètre un tableau de tuplets (Nom, Prénom) des nouveaux clients et renvoie un tableau de chaînes de caractères. Par exemple, `formatage(tab_clients)` renvoie :
['LOVELACE_Ada', 'BOOLE_George', 'VONNEUMANN_John', 'SHANNON_Claude', 'KNUTH_Donald']
- 4) Écrire une fonction `creation_dossiers(tab)` qui prend en paramètre un tableau de chaînes de caractères et qui crée et modifie les droits des dossiers au nom de ces chaînes de caractères avec les mêmes droits que le sous-dossier TURING_Alan.