

TP5 - récapitulatif

Exercice 1 : Les filtres et les redirections

1. Créer dans le répertoire d'accueil les fichiers **toto.txt** et **tutu.txt**.
2. Créer un fichier **exo.txt** dont le contenu est le suivant

ligne01 Début du fichier

Ligne 02 Deuxième ligne

ligne3 il ne faut pas traiter cette ligne

ligne 4 je dois apparaître dans les résultats :

ligne05 ,:/rep13/ moi aussi

ligne 06 pas moi

ligne07 moi dans deux questions :

_ligne8 une erreur de frappe ?

Eligne9 ,:/non d/

c'est un joli fichier que nous avons là

Avec des majuscules et des minuscules en premier

et des numéros de lignes

ainsi que des :

et des /

3. Écrire une commande qui enregistre dans le fichier **toto.txt** toutes les lignes du fichier **exo.txt** qui contiennent le mot « **ligne** » suivi immédiatement de deux chiffres.
4. Écrire une commande qui enregistre dans le fichier **tutu.txt** toutes les lignes du fichier **exo.txt** qui contiennent le symbole « : » en fin de ligne.
5. Écrire une commande qui enregistre à la suite du contenu du fichier **tutu.txt**, toutes les lignes du fichier **exo.txt** qui contiennent une virgule, suivie du symbole « : », suivi du symbole « / », suivi de cinq caractères quelconques, suivi à nouveau du symbole « / ».
6. Écrire une commande qui affiche le nombre de lignes du fichier **exo.txt** qui commencent par un caractère qui ne soit pas un tiret bas « _ » ni la lettre **E**.
7. Écrire une commande qui affiche toutes les lignes du fichier **exo.txt** qui commencent par une lettre minuscule de l'alphabet, triées dans l'ordre alphabétique inverse.

Exercice 2 : Script 1

Créer un script qui doit calculer et afficher le nombre de fichier(s) ordinaire(s) et le nombre de sous-répertoire(s) d'un répertoire quelconque qui sera donné en paramètre (ou saisi en cas d'absence du paramètre). Le script devra afficher une erreur si le répertoire à analyser n'existe pas ou ne correspond pas à un répertoire exécutable ou avec les droits de lecture.

Exercice 3 : Script 2

Écrire un script qui permet la saisie d'un identifiant d'utilisateur et qui affiche si cet utilisateur existe et qui, en cas de réponse positive, affiche également l'**UID** correspondant. Le programme devra boucler en redemandant un nouvel utilisateur à chaque fois jusqu'à ce que l'on décide de quitter en tapant la lettre « **q** ».

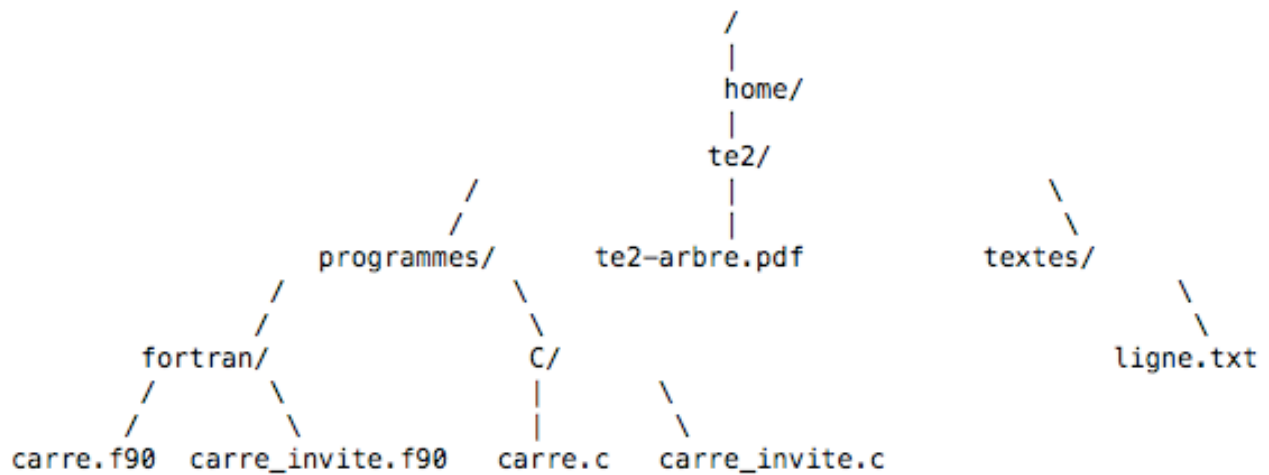
Pour savoir si un utilisateur existe, on peut se servir du fichier **/etc/passwd** dans lequel chaque utilisateur correspond à une ligne qui a la structure suivante :

Identifiant:x:UID:GID:nom:répertoire d'accueil:shell

Exercice 4: Hiérarchie des fichiers et commandes de base

Dans l'exemple ci-dessous on considère que l'utilisateur connecté est « **te2** ». Il faudra adapter cet exemple au nom de l'utilisateur courant.

1. Créer l'arborescence de l'exemple suivant.



2. Comment obtenir la liste des fichiers se terminant par l'extension **.c** du répertoire **C** avec leurs droits associés en utilisant un chemin absolu ?

3. La commande **ls -a ~/programmes** donne le résultat suivant :

```

.
..
fortran

```

A quoi correspond **.** et à quoi correspond **..** ?

4. Créer un répertoire **doc** dans le répertoire **programmes**.

5. Effectuer un déplacement du répertoire courant vers le répertoire **textes**. Déplacez le répertoire **doc** dans le répertoire **textes** en utilisant des chemins relatifs (vous ne devez donc pas utiliser **~**).

6. Rendre exécutable tous les fichiers contenus dans le répertoire **C**.

7. Faire en sorte que les fichiers du répertoire **fortran** ne soient lisibles que pour l'utilisateur et que personne ne puisse ni modifier ni exécuter ces fichiers.

8. Renommer le fichier **ligne.txt** en **toto.txt**.

Exercice 5: Les processus

1. Visualiser la liste de tous les processus de **root** tournant sur la machine.

2. Suspendre un processus lancé au premier plan dans le terminal courant, puis le relancer en arrière plan.

3. Visualiser uniquement les processus lancés depuis le terminal courant.

4. Arrêter tous les processus ayant pour nom **top**.