**Exercice 02 – Cours 04  
Opérations sur les fichiers et répertoires**

**Partie 1  
Outils de recherche dans le système de fichiers**

Sous Linux, les fichiers sont utilisés pour stocker des données telles que du texte, des images, de vidéo ou des programmes.   
Les répertoires sont eux aussi considérés comme des fichiers qui contiennent d'autres fichiers.

Dans cet exercice, nous verrons comment rechercher des fichiers dans le système de fichiers hiérarchique de Linux (FHS).

Les commandes **locate** et **find** ont toutes deux le même rôle:   
elles sont utilisées pour lancer une recherche dans le système de fichiers.

Bien que ces deux commandes exécutent des tâches similaires, chacune le fait en utilisant une technique différente avec ses propres avantages et inconvénients.

La commande **locate** utilise une base de données normalement mise à jour une fois par jour à l'aide d'une tâche planifiée tandis que la commande **find** recherche en parcourant le système de fichiers à chaque appel.

**Étape 01**La commande **locate** pourrait ne pas être installée sur votre système …  
**>> locate  
-bash: locate: command not found**

Pour l'installer, on utilise le gestionnaire de paquets de notre distribution.  
**>> sudo apt update && sudo apt upgrade --yes  
>> sudo apt search -n locate**ou **>> sudo apt list locate****>> sudo apt install locate**

**Étape 02**La commande **locate** utilise une base de données contenant l'emplacement des fichiers sur le système de fichiers.   
Cette base de données doit être systématiquement mise à jour quotidiennement ou avant de lancer une nouvelle recherche.

La commande **locate** accepte une chaîne de caractères comme argument …  
**>> locate [Option(s)] ... <Motif> …**

Par suite de l'installation de **locate**, sa base de données est vide.   
N'importe quelle recherche retourne un résultat vide …  
**>> locate hostbane**

**Étape 03**Avant d'utiliser la commande **locate**, il convient de mettre à jour sa base de données à l'aide de la commande **updatedb**. Cette dernière doit être lancée par un utilisateur qui dispose des privilèges du root.

La mise à jour de la base de données est souvent une tâche automatisée par le système.   
L'utilisateur n'a pas à se soucier de la mettre à jour lui-même à moins que le système de fichiers ait subi des modifications depuis sa dernière indexation.

Pour mettre à jour la base de données …  
**>> sudo updatedb**

**Partie 2  
Extraction d‘informations dans les fichiers**

Le texte est la principale forme d'entrée/sortie des commandes Linux.   
Par conséquent, il existe de nombreuses commandes dont le rôle est de traiter des fichiers texte.

**Étape 01 (cat)**La commande cat (dérivée du mot concaténer) prend plusieurs fichiers en entrée et les affiche bout à bout.

Sa syntaxe est la suivante …  
**>> cat [Option(s)] ... <Fichier(s)> …**

Pour afficher le contenu des fichiers **/etc/hosts** et **/etc/hostname** …  
**>> cat /etc/hosts**  
**>> cat /etc/hostname**

Pour concaténer (afficher bout à bout) le contenu des fichiers **/etc/hosts** et **/etc/hostname** …  
**>> cat /etc/hosts /etc/hostname**

**Étape 02**Bien que l'affichage de petits fichiers avec la commande **cat** est couramment utilisée, ce n'est pas la commande idéale pour visualiser des fichiers volumineux dont le nombre de lignes dépasse ce qui peut être affiché d'un coup à l'écran.

Pour les fichiers plus volumineux, une commande de pagination peut être utilisée pour afficher le contenu long.

Les commandes de pagination affichent le texte page par page, ce qui vous permet d'avancer et de reculer dans le fichier à l'aide des touches de déplacement.

La commande **less** est l'outil de pagination par défaut de plusieurs distributions Linux.   
Nous l'avons d'ailleurs déjà rencontré plus tôt dans ce cours avec la commande man.   
La commande man affiche le manuel d'aide d'une commande page par page.

Pour afficher un fichier avec la commande **less**, on fournit le nom du fichier en argument …  
**>> cat /etc/passwd**

La touche espace permet de défiler les pages de **less** jusqu'à la fin.   
Pour quitter le manuel, appuyez sur **q**.

**Étape 03**Expérimentez maintenant les différents déplacements possibles dans **less**.

|  |  |
| --- | --- |
| Commande | Rôle |
| Entrée | Descendre d'une ligne |
| Espace | Descendre d'une page |
| /qqc | Rechercher "qqc" dans la page |
| n | Repérer le prochain élément recherché |
| 1G | Retourner au début du fichier |
| G | Aller à la fin du fichier |
| h | Afficher l'aide |
| q | Quitter less |

**Étape 04**La commande **head** permet d'afficher le début d'un fichier ou d'une sortie à l'écran …  
**>> head [Option(s)] ... <Fichier> …**

Pour afficher les 10 premières lignes du fichier **/etc/passwd** …  
**>> head /etc/passwd**

Si le nombre de lignes n'est pas spécifié, la commande **head** affiche les **10 premières lignes**.   
Pour afficher les **5 premières ligne**s de la sortie du fichier précédent …  
**>> head --lines 5 /etc/passwd  
>> head -n 5 /etc/passwd**  
**>> head -5 /etc/passwd**

**Étape 05**  
La commande **tail** affiche les dernières lignes d'un fichier …  
**>> tail [Option(s)] ... <Fichier> …**

Pour afficher les 10 dernières lignes du fichier **/etc/passwd** …  
**>> tail /etc/passwd**

Si le nombre de lignes n'est pas spécifié, la commande **tail** affiche les **10 dernières lignes**.   
Pour afficher les **5 dernières ligne**s de la sortie du fichier précédent …  
**>> tail --lines 5 /etc/passwd  
>> tail -n 5 /etc/passwd**  
**>> tail -5 /etc/passwd**

**Étape 06**La commande **cut** est utilisée pour extraire des champs d'un fichier texte …  
**>> cut [Option(s)] ... <Fichier> …**

L'espace et la tabulation sont les délimiteurs (ce qui sépare les champs) par défaut et peuvent être modifiés à l'aide du commutateur **--delimiter** (forme courte **-d**).

Le commutateur **--fields** permet de préciser les colonnes à afficher.

Pour extraire les 1er, 3e et 7e champs du fichier **/etc/passwd** …  
**>> cut --delimiter ':' --field '1,3,7' /etc/passwd**ou  
**>> cut -d ':' -f 1,3,7 /etc/passwd**

**Étape 07**La commande **sort** est utile pour travailler avec des données structurées avec des colonnes.  
Elle permet d'afficher un fichier trié sur un champ de données spécifique …  
**>> cut [Option(s)] ... <Fichier> …**

Par défaut le délimiteur entre les champs (colonnes) est l'espace.  
Il peut être spécifié par le commutateur **--field-separator** (forme courte **-t**).   
Le champ sur laquelle on souhaite trier est donné par le commutateur **--key** (forme courte **-k**).

Pour trier le fichier **/etc/passwd** en fonction du nom des utilisateurs présents sur le système (1ère colonne …  
**>> cat /etc/passwd   
>> sort --field-separator ':' --key '1' /etc/passwd**ou  
**>> sort -t ':' -k '1' /etc/passwd**

**Remarque** …  
Le délimiteur entre les colonnes est les deux-points (:).

**Étape 08**  
La commande grep agit comme un filtre.   
Elle permet d'extraire les lignes qui contiennent une chaîne de caractère fournie par l'utilisateur.  
Cette commande est très utile pour extraire une donnée spécifique dans un fichier complexe …  
**>> grep [Option(s)] ... <Motif(s)> <Fichier> …**

Pour extraire toutes les lignes du fichier **/etc/passwd** qui contiennent la chaîne root …  
**>> grep root /etc/passwd**

La commande grep est sensible à la casse.  
Pour extraire toutes les lignes du fichier **/etc/passwd** qui contiennent la chaîne root (en minuscule ou majuscule) …  
**>> grep --ignore-case root /etc/passwd**ou  
**>> grep -i root /etc/passwd**

**Étape 09**Il est commun qu'un utilisateur souhaite paginer la sortie d'une commande très longue.   
Un exemple concret pourrait être de lister au format long le contenu du répertoire **/etc**.

Lancez la commande **ls -l /etc**.  
On remarque que la sortie de la commande est plus longue que la taille de notre écran et que les premières lignes du la sortie sont passées sous nos yeux sans que l'on ait pu les lire.

Il existe deux solutions à ce problème.

* Premièrement utiliser la roulette de la souris pour remonter dans la sortie du terminal virtuel.   
  Ceci est possible seulement dans un terminal virtuel en environnement graphique et le nombre de lignes conservées est limité.
* La seconde et meilleure solution consiste à paginer la sortie de la commande pour l'afficher page par page.   
  Pour ce faire, on enchaîne la commande souhaitée à **less** en ajoutant **| less** (prononcer pipe less, à l'anglaise).

Par exemple, la commande **ls -l /etc** deviendrait **ls -l /etc | less**.   
Le résultat de la commande **ls** devient ainsi l'entrée de la commande **less**.

C'est un exemple très commun d'enchaînement de commandes sous Linux.

On peut ensuite faire défiler l'affichage en utilisant les touches de navigation de la commande **less** comme espace pour défiler les pages …  
**>> ls -l /etc | less**

On peut également reprendre les commandes cut et sort tout en créant un tuyau (pipeline) …  
**>> cut --delimiter ':' --field '1,3,7' /etc/passwd | sort --field-separator ':' --key '1'**ou  
**>> cut -d ':' -f 1,3,7 /etc/passwd | sort -t ':' -k '1'**

Il est même possible de filtrer le résultat à l’aide de la commande grep …  
**>> cut --delimiter ':' --field '1,3,7' /etc/passwd | sort --field-separator ':' --key '1' | grep --ignore-case root**ou  
**>> cut -d ':' -f 1,3,7 /etc/passwd | sort -t ':' -k '1' | grep -i root**

**Étape 10**Toujours en utilisant le principe d'enchaînement, nous pouvons compter le nombre de fichiers que contient le répertoire **/etc**.

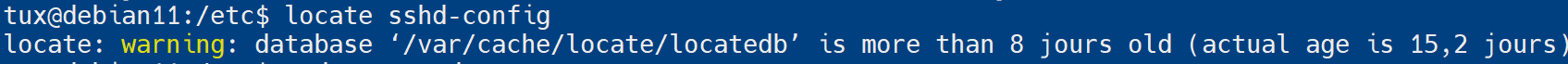
Sachant que la commande **ls -l /etc** produit une ligne par fichier à la sortie, on peut enchaîner cette commande à **wc -l** qui compte le nombre de lignes de son entrée …  
**>> ls -l /etc/ | wc --lines**ou  
**>> ls -l /etc/ | wc -l**

**Remarque** …  
Le commutateur **--lines** (forme courte **-l**) avec la commande **ls** ne signifie pas la même chose que le commutateur **-l** avec la commande **wc**.   
On pourrait réécrire une commande équivalente avec les options longues, ce qui donne …  
**>> ls --format=long /etc/ | wc --lines**

**Questions de révision et d'expérimentation**

**Question 01**

* Utilisez la commande **locate** pour localiser un fichier nommé **auth.log**.  
  **--------------**
* Utilisez la commande **find** pour localiser un fichier nommé **sshd\_config**.  
  **--------------**
* Utilisez la commande **find** pour localiser tous les fichiers portant le **suffixe .txt**.  
  **--------------**

**Remarque** …  
Ne pas oublier de mettre à jour l’index de la base de données de l’utilitaire **locate** …  
  


**Question 02**

Créez le fichier **~/fruits.liste** avec le contenu suivant …  
  
Une image contenant texte

Description générée automatiquement

Utilisez le fichier **~/fruits.liste** pour répondre aux questions suivantes.

* Affichez les lignes qui contiennent les chaînes **fraise** et **cerises**.  
  **--------------**
* Utilisez l'enchaînement de commandes.   
  On retrouve combien de mots, de lignes et de caractères dans le fichier **~/fruits.liste** ?  
  **--------------  
  --------------  
  --------------**

**Question 03**

* Considérant le fichier **/etc/passwd**.   
  Affichez tous les noms d'utilisateurs qui ont l’interpréteur de commande (*shell*) de connexion /**bin/bash**.  
  Il s’agit du dernier champ sur chaque ligne du fichier **/etc/passwd**.   
  Affichez seulement leur nom.  
  **--------------**
* Toujours à partir du fichier **/etc/passwd …**Affichez le nombre d'utilisateurs du système qui ont l’interpréteur de commande (*shell*) de connexion **/usr/sbin/nologin**.  
  **--------------**

Avec la collaboration de **Miguel Grandmont Champagne**.