**Ce laboratoire permet de mieux comprendre plusieurs commandes de Linux pour la gestion des fichiers comme grep, find, sort, cut plus les commandes alias, unalisas. Le dernier exercice est au sujet des liens physiques et des liens symboliques.**

**Décrire ce que fait chacune des commandes "grep"**

1. grep "^.o" /etc/passwd

**cherche les lignes dont la première lettre est quelconque et la 2ème doit être o**

réponse:

1. grep "^r" /etc/passwd

**cherche les lignes commençant par la lettre r**

réponse:

1. grep -v "^r" /etc/passwd

**cherche les lignes ne commençant pas commençant par la lettre r**

réponse:

1. grep "r..t" /etc/passwd

**cherche les lignes contenant les mots qui respectent le modèle r..t**

réponse:

1. grep -c "r..t" /etc/passwd

**affiche le nombre de lignes contenant les mots qui respectent le modèle r..t**

réponse:

1. grep [Ay] /etc/passwd

**cherche les lignes qui contiennent A ou y**

réponse:

1. grep –n "^r'' /etc/passwd\*

**cherche et numérote les lignes qui débutent avec r dans les fichiers /etc/passwd\***

réponse:

1. ls -l /etc/ | grep "^d"

**permet de lister les sous-répertoires du répertoire /etc/**

réponse:

**Décrire ce que fait chacune des commandes "find"**

1. find / -iname passwd

**cherche les fichiers dont le nom est passwd à partir de la racine**

**-iname ignore les majuscules et les minuscules**

réponse:

1. find / -size +1000000c

**cherche les fichiers qui occupent plus de 1 000 000 d'octets**

réponse:

1. find / -size +1000000c -and -name b\*

**cherche les fichiers qui occupe plus de 1 000 000 d'octets et dont le nom débute par b**

réponse:

1. Écrivez la commande pour trouver à partir de la racine les répertoires dont le nom débute par x11 et sans tenir compte des majuscules et des minuscules ?

**important: le résultat de votre commande doit afficher seulement les répertoires**

**find / -iname x11\* -type d**

réponse:

1. Écrivez la commande pour trouver à partir de la racine les fichiers dont le nom débute par x11 et sans tenir compte des majuscules et des minuscules ?

**important: le résultat de votre commande doit afficher seulement les fichiers**

**find / -iname x11\* -type f**

réponse:

**Décrire ce que fait chacune des commandes "cut"**

1. cat /etc/passwd | cut -d: -f1

**affiche le premier champs du fichier /etc/passwd**

réponse:

1. cat /etc/passwd | cut -c1-10

**affiche les caractères qui sont dans les colonnes 1 à 10**

réponse:

**Décrire ce que fait chacune des commandes "alias" et "unalias"**

1. alias

**Affiche la liste des alias**

réponse:

1. Prendre en note l'alias ls

**ls --color=auto**

réponse: ls=

1. Création de trois alias

alias cls=clear **(efface le contenu d'un terminal)**

alias clearall="echo -en '\033c\033[3J'" **(efface complètement le contenu d'un terminal)**

alias lla="ls -la"

1. alias

**Affiche la liste des alias, vous devriez voir les trois nouveaux alias.**

réponse:

1. Élimination de trois alias

unalias cls

unalias clearall

unalias lla

1. alias

**Affiche la liste des alias, vous ne devriez plus voir les trois alias**

réponse:

1. cat /home/tech/.bashrc

En utilisant l'éditeur de fichier **nano**,ajouter les alias **cls** et **clearall** dans le fichier **/home/tech/.bashrc**

Pour exécuter le fichier qu'on vient de modifier on doit exécuter la commande suivante:

**source /home/tech/.bashrc**

**Décrire ce que fait chacune des commandes "sort"**

note: vous devez récupérer le fichier info.txt qui est disponible sur LÉA.

note: vous pouvez utiliser WinSCP.exe pour copier le fichier vers le dossier /tmp/ de votre LINUX

1. clear; cat info.txt ; echo ----------; sort -t: -r info.txt

**affiche le fichier. Affiche le fichier trié en ordre décroissant sur le premier champ**

réponse: -t : le deux point c est pour separere les valeurs on pourrait mettre . ou ; ou -

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. clear; cat info.txt ; echo ----------; sort -t: -k 2 info.txt

**affiche le fichier. Affiche le fichier trié sur le deuxième champ**

réponse: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. clear; cat info.txt ; echo ----------; sort -t: -k 3.2 info.txt

**affiche le fichier. Affiche le fichier trié par le 2ème caractère du troisième champ**

réponse: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. clear; cat info.txt ; echo ----------; sort -t: -n -k 4 info.txt

**affiche le fichier. Affiche le fichier trié par la 4ème champ et considérer son contenu comme un nombre**

réponse: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### La commande sort -t: -n -k 4 info.txt utilise la commande sort pour trier les lignes d'un fichier texte (info.txt) en fonction de la quatrième colonne (champ) délimitée par le caractère :.

### Explications des options utilisées dans la commande :

### sort: La commande de tri.

### -t: : Spécifie le délimiteur de champ, ici, le caractère :. Cela indique à sort que les colonnes sont séparées par des deux-points.

### -n: Effectue un tri numérique plutôt qu'un tri lexicographique. Cela est utile lorsque la quatrième colonne contient des valeurs numériques.

### -k 4: Indique que le tri doit être effectué sur la quatrième colonne du fichier.

1. clear; sort -t: info.txt -o resultat.txt

**trie le fichier info.txt et place le résultat dans le fichier resultat.txt**

réponse: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Exercices sur les liens physiques**

cd /tmp

ls -l > test.txt

ln test.txt lien.txt

ls -li test.txt lien.txt

**65 -rw-r--r-- 2 root root 607 Sep 9 19:12 lien.txt**

**65 -rw-r--r-- 2 root root 607 Sep 9 19:12 test.txt**

Que constatez-vous ? (2 observations)

**Les deux fichiers ne sont qu'un seul fichier avec deux noms. Ils ont le même inode.**

réponse: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Le chiffre 2 indique le nombre de fichiers qui utilisent le même inode.**

réponse: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Un inode est une structure de données fondamentale utilisée pour stocker des informations (nom, taille du fichier, autorisations, nombre de liens, …) sur un fichier ou un répertoire.**

**Exercices sur les liens symboliques**

cd /tmp

ls -l > test.txt

ln -s test.txt tata.txt

ls -li test.txt tata.txt

**655824 lrwxrwxrwx 1 root root 8 Sep 8 19:12 tata.txt -> test.txt**

**655823 -rw-r--r-- 1 root root 607 Sep 8 19:12 test.txt**

Que constatez-vous ? (3 observations)

**La lettre "l" indique que nous avons un lien symbolique.**

réponse:

**La taille du fichier tata.txt est très petite par rapport au fichier test.txt.**

réponse:

**Le fichier tata.txt pointe vers le fichier test.txt.**

réponse: