

LAB2 – Traitement des chaînes de caractères, pipes, filtres, et redirections

Objectif

Le but de ce Lab est de maîtriser les commandes relatives à la manipulation des chaînes de caractères, la redirection des Entrées/Sorties standards, et l'utilisation des "pipes".

Partie 1 : Traitement des chaînes de caractères et redirection des Entrées/Sorties standards

1. Copier le fichier `/etc/passwd` dans votre répertoire personnel, sous le nom `mypasswd`.
2. Visualiser le contenu du nouveau fichier créé `mypasswd` à l'aide des commandes `cat`, `less`, et `more`, et vérifier que chaque ligne de ce fichier décrit la configuration d'un compte utilisateur sur le système.
3. Visualiser le contenu du fichier `mypasswd` dans l'ordre inverse des lignes, à l'aide de la commande `tac`.
4. À l'aide des commandes `head` et `tail`, afficher respectivement les 5 premières lignes et les 5 dernières lignes du fichier `mypasswd`.
5. En utilisant la commande `wc`, compter le nombre de comptes utilisateurs existants sur la machine.
6. À l'aide de la commande `nl` afficher le contenu du fichier `mypasswd` en numérotant chaque ligne.
7. Générer une liste des différents shells en cours d'utilisation sur l'ordinateur et l'enregistrer dans un fichier sous le nom `shells`. Visualiser par la suite le contenu du nouveau fichier créé `shells`. Expliquer le rôle des options `-d` et `-f`.
8. Que permettent ces deux commandes de réaliser ?
`$ cut -c 1-10 mypasswd > mypasswd1`
`$ cut -c 11- mypasswd > mypasswd2`
9. Fusionner le contenu des deux fichiers `mypasswd1` et `mypasswd2` à l'aide de la commande `paste`.
10. Le fichier `shells` contient toutes les informations recherchées, mais sans un ordre particulier. Trier les lignes de la sortie et placer les nouvelles données dans un nouveau fichier intitulé **sorted.shells**.
11. Certaines valeurs identiques se répètent dans le fichier **sorted.shells**. Utiliser la commande `uniq` pour déterminer le nombre d'instances de chaque valeur et enregistrer le résultat dans un fichier `uniq.sorted.shells`.
Pourquoi il est nécessaire de trier la sortie avant d'utiliser la commande `uniq` ? Vous pouvez essayer la commande :
`$ uniq -c shells`
12. Utiliser la commande `sort` pour obtenir une liste triée dans l'ordre décroissant des **shells** en cours d'utilisation sur l'ordinateur, où chaque **shell** affiché sera précédé par le nombre d'occurrences dans le fichier.
13. Copier le fichier `/usr/share/dict/words` dans votre répertoire personnel, sous le nom **mywords**.

14. Exécuter la commande suivante pour obtenir en sortie les lignes du fichier **mywords** (se trouvant dans votre répertoire personnel) contenant un motif de texte précis (par exemple le motif **fish**) :
`$ grep fish mywords`
Comparer le résultat obtenu avec la commande : `$ grep -i fish mywords`
15. Utiliser la commande **grep** pour afficher toutes les lignes du fichier **mywords** contenant le motif **fish**, de même que les deux lignes succédant et précédant chacune de ces lignes (afin d'obtenir un contexte plus étendu)
16. Utiliser la commande **grep** pour répondre à chacune des questions suivantes :
 - a. Indiquer le nombre de fois que le motif **fish** apparaît dans le fichier **mywords**.
17. En utilisant le fichier déjà créé **mypasswd**, réaliser les tâches suivantes :
 - c. Afficher la ligne de tout compte utilisateur commençant par la lettre **g**.
 - d. Afficher la ligne de tout compte utilisateur utilisant le shell **/bin/bash**.
 - e. Afficher la ligne de tout compte n'utilisant pas le shell **/bin/bash**.
18. Créer un fichier **modified.passwd** à partir du fichier **mypasswd** ne contenant aucune ligne ayant les lettres **N** ou **P**:
18. Utiliser la commande **tr** pour convertir toutes les lettres majuscules (s'il y en a) en minuscules :
19. Utiliser **cat** pour visualiser le fichier **mypasswd** et le nouveau fichier **modified2.passwd**.

Partie 2 : Utilisation des pipes

22. Pour créer une liste triée de **shell**, les pipes seront utilisées pour combiner certaines commandes exécutées dans les questions précédentes. Ainsi, il n'y a plus besoin d'aucun fichier temporaire car l'exécution est faite en conduisant la sortie d'une commande vers une autre :
`$ cut -d: -f7 mypasswd | sort | uniq -c | sort -nr`
23. Utiliser la commande **ps** pour obtenir une liste des processus en cours d'exécution ainsi que le nom d'utilisateur ayant lancé l'exécution de chacun d'entre eux. Puis conduire la sortie vers **grep** pour n'obtenir que la sortie des processus appartenant à l'utilisateur **root**. Le symbole **^** indique à **grep** de ne chercher que la chaîne **root** au début de chaque ligne : `$ ps auxx | grep ^root`
24. Imaginer qu'il faut passer ces données à un autre programme ou script qui n'a besoin que des noms des processus. Utiliser une combinaison des commandes **tr** et **cut** pour produire cette liste. Premièrement **tr** est utilisée pour transformer les espaces blancs en un séparateur de champs pouvant être compris par **cut** (un **%** dans ce cas). Ensuite, **cut** est utilisé pour afficher uniquement les noms de processus (champ **11** de la sortie de la commande **tr**) :
`$ ps auxx | grep ^root | tr -s [:blank:] [%] | cut -d% -f11`
25. Vous avez maintenant la liste des processus appartenant à **root**, mais comme vous avez demandé à **cut** de n'afficher que le champ **11**, vous avez perdu les arguments de la ligne de commande. Si vous changez **-f11** par **-f11-**, **cut** affichera alors le champ **11** jusqu'à la fin de la ligne.
`$ ps auxx | grep ^root | tr -s [:blank:] [%] | cut -d% -f11-`
26. Dans la question précédente la commande **tr** a été utilisée pour transformer les espaces blancs en **%** pour que la sortie de la commande puisse être utilisée par la commande **cut**. Utiliser la commande **tr** de nouveau pour remettre les **%** en espaces blancs.
`$ ps auxx | grep ^root | tr -s [:blank:] [%] | cut -d% -f11- | tr [%] " []"`

Partie 4 : Utilisation de la commande sed

28. A l'aide de la commande **nano**, créer un fichier exemple ayant le contenu suivant (n'oublier pas d'inclure les sauts de lignes) :

```
Bienvenue,  
  
C'est le fichier  
exemple. Contenu de la  
ligne 4.  
  
# ceci est un commentaire  
Contenu de la ligne 7.  
  
A la prochaine
```

29. A l'aide de la commande **sed** (qui représente un utilitaire de traitement de données capable d'utiliser les expressions régulières) effacer le contenu des lignes 4 et 7 du fichier exemple.
30. Supprimer le contenu des lignes commentaires commençant par un dièse (les deux caractères slash permettent d'inclure une expression régulière) :
31. Afficher uniquement les lignes contenant la chaîne « Contenu » :
32. Remplacer la chaîne de caractère # par * dans le contenu du texte :

Partie 5 : Recherche de fichiers

33. A l'aide de la commande **find**, qui permet de chercher des fichiers, réaliser les tâches suivantes :
- Afficher la liste des fichiers se trouvant dans le répertoire **/etc** et dont le nom se termine par **.conf**, en invoquant la commande avec le paramètre **-name**.
 - Retrouver la liste des fichiers spéciaux de type bloc, et de type caractère, ainsi que la liste des fichiers standards et des répertoires, se trouvant dans le répertoire **/dev**, en invoquant la commande avec le paramètre **-type**.
 - Retrouver la liste de tous les fichiers dont la taille dépasse 10Mo, en invoquant la commande avec le paramètre **-size**.
 - Retrouver la liste des fichiers modifiés durant les dernières 24 heures, en invoquant la commande avec le paramètre **-mtime**.
 - Retrouver la liste des fichiers vous appartenant, en invoquant la commande avec le paramètre **-user**.
34. Une autre alternative à l'utilisation de **find**, est la commande **locate**. Cette dernière est plus rapide car elle repose sur une recherche dans la base de données **mlocate**.
- Commencer par mettre à jour cette base de données avec la commande **\$updatedb**.
 - A l'aide de la commande **locate**, chercher la liste des fichiers dont le nom contient la chaîne de caractères **passwd**.
 - A l'aide de la commande **locate**, afficher le nombre de fichiers dont le nom contient la chaîne de caractères **passwd**.