MANICOME
Saravanane

COMPTE RENDU DU TP 2

Exercice 1

1. find / -name passwd >> lst_passwds.log 2>> lst_passwds.err

2. ls -Rl >> sauvegarde.txt

3. wc -l sauvegarde.txt
   564

4. sort sauvegarde.txt | uniq sauvegarde.txt | wc -l
   564, il n'y a pas de fichier en double

5. ls -l | wc -l

6. ls -Rl / | egrep passwd | wc -l

7. find / -name passwd 2>/dev/null | cat > passwd_all

8. La commande tee permet d'écrire dans un ou plusieurs fichier dont la sortie standart depuis l'invite de commande.

9.

10. Cette commande écrit dans le fichier ~/mon_rep le contenu du répertoire personnel

11. Cette commande écrit dans le fichier ps.txt les processus en cours en redirigeant les erreurs vers la sortie standart, qui sont donc écrit dans le fichier également.

Exercice 2

1. La commande ls permet d’afficher les entrées du répertoire concerné.

2. ls /usr/include

3. ls /usr/include | grep "^c"

4. ls /usr/include | grep "i"

5. ls /usr/include | grep ".*i.*\.h"

6. ls /usr/include | grep ".*std.*"

7. ls /usr/include | grep "^...$"

8.
   a) Cette commande supprime les fichiers foo415.txt et foo416.txt en demandant confirmation.

   b) Cette commande supprime soit le fichiers foo415.txt soit le fichier 416.txt en demandant confirmation.

9. rm -i !(?(foo(.*).txt))
   Supprime tous les fichiers foo*.txt présent plus d’une fois.

   rm -i foo@(.*).txt
   Supprime tous les fichiers foo*.txt uniques

   rm -i @(foo(.*)).txt
   Supprime un fichier parmis une liste de foo(...).txt, s’il y a 10 fichier foo1.txt cette commande n’en supprimera qu’un, idem pour foo2.txt, foo3.txt…

10.
   a) echo &
      [1] 8874

   b) echo #

   c) echo ()
      >

   d) echo |
      >

e) echo \
   >

f) echo ^
   ^

g) echo @
   @

h) echo $
   $

i) echo *
   `liste les fichiers du répertoire`

j) echo !
   !

11.
   a) echo $HOME → Affiche le répertoire personnel
   b) echo ‘$HOME’ → Affiche $HOME
   c) echo «$HOME» → Affiche le répertoire personnel et le reste du texte concaténé
   d) echo \$HOME → Affiche $HOME

12. echo "Bonjour, je suis `whoami`, nous sommes le `date +'%A %d %B %Y %H heures %M minutes %S secondes'`"

13. La commande permet d’afficher que le nom des processus s’exécutant sur la machine local et pas sur la machine distante.

14. L’entrée de la commande grep est la sortie standart de la commande ps avec une redirection des erreurs sur la sortie standart.

Exercice 3

1.
2. grep "^`whoami | cut -c 1`" /etc/passwd
3. ps | grep "pts"
4. grep "bash" /etc/passwd
5. grep "^.*/bash$" /etc/passwd && grep "^.*/sh$" /etc/passwd
6. ls -l ~ | grep "rwx------"
7. ls -l ~ | egrep '^(d|-)'
8. ls -a ~ | egrep '^\.'
9. ps | egrep 'tty|pts' | tee result.txt
10. var=`grep 'UID' /etc/passwd | cut -d : -f1`
11. head -n 10 /etc/passwd | tail -n +3
12. head -n 10 /etc/passwd | tail -n +3 | nl
13. ps | wc -l
14. cat /etc/passwd | sort -n -t ':' -k3
15.
    a) split -l n file

       → Permet de séparer un fichier en plusieurs autres de n ligne maximum
    b) split -b n file

       → Permet de séparer un fichier en plusieurs autres de taille n maximum
    c) split -n n file

       → Permet de séparer un fichier en n autres fichiers
16.
    a) sudo wc -l /etc/passwd /etc/shadow

       →82 lignes, 41 dans chaque
    b) cd ~ | mkdir temp
    c) cp /etc/passwd temp

       sudo cp /etc/shadow temp
    d) cut -d ':' -f 1 passwd > passwd.users

       cut -d ':' -f 1 shadow > shadow.users

e) wc -l passwd.users shadow.users

f) sort -d passwd.users | uniq > passwd.users.s

   sort -d shadow.users | uniq > swhado.users.s

g) diff passwd.users.s shadow.users.s

   → aucune différence n'esst affichée

h) sort -n -t ':' -k3 passwd > passwd.sUID

i) head -n 19 /etc/passwd | tail -n +10 | sort -d -r > sub.txt

j) cat passwd | tr 'aeiouy\n\ 0' '123456\!_\.'

k) Cette commande retourne l'espace occupé par tous le fichiers se terminant par .conf dans le repertoire d'execution.