

Systemes d'Exploitation

TP 1

➤ Exercice 1

Lit le fichier `/etc/passwd`. Considère un numéro de groupe parmi ceux existants et affiche le nombre de personnes appartenant a ce groupe.

```
grep 570 /etc/passwd | wc -l
```

➤ Exercice 2

Crée trois fichiers contenant le nom de certains fichiers du répertoire `/etc` dont les caractéristiques sont :

- fich1 : noms des fichiers tels que `taille < 2000`
- fich2 : noms des fichiers tels que `2001 <= taille < 4000`
- fich3 : noms des fichiers tels que `4001 <= taille`

La taille est en nombre de caractères. Crée ensuite un répertoire `taille` puis transfère ces fichiers dans ce répertoire.

```
find /etc -size -2000c -exec ll -d + > fich1
find /etc -size +2001c -size -4000c -exec ll -d + > fich2
find /etc -size +4001c -exec ll -d + > fich3
mkdir ~/taille
mv fich1 taille
mv fich2 taille
mv fich3 taille
```

➤ Exercice 3

Crée un fichier vide appelé « vide » (original, non ?) dans le répertoire `taille`, puis change ses droits pour qu'il soit exécutable par l'utilisateur uniquement.

Puis on transfère le fichier à son binôme, en le copiant chez lui et en le rendant propriétaire du fichier.

```
touch vide
chmod 766 vide
cp vide ~/../binome
chown binome ~/../binome/vide
```

Systèmes d'Exploitation

TP ②

➤ Exercice 1

Modification du fichier .cshrc pour qu'à chaque connexion, l'utilisateur aie des indications sur les utilisateurs connectés, ainsi que la date et l'heure de connexion :

```
echo "\n> Utilisateurs connectes :"  
who | cut -d " " -f1 | sort | uniq | tr "\n" " "  
echo "\n"  
date "+> Bienvenue, il est %H:%M:%S%n Nous sommes le %A %d %B %Y"  
echo ''
```

➤ Exercice 2

Script permettant de vérifier si un utilisateur est connecté ou pas. Ce script prendra en argument le nom de l'utilisateur, et renverra un message pour indiquer la connexion ou non de l'utilisateur. De plus, le programme devra reconnaître si un paramètre a été transmis ou non.

```
if [ -n "$1" ]  
then  
    if [ "$1" = "people" ]  
    then  
        echo "Nokia : Connected People"  
    else  
        i=`who | cut -d " " -f1 | sort | uniq | grep -c "^$1$"`  
        if test $i -gt 0  
        then echo "$1 est connecte !"  
        else echo "$1 n'est pas connecte..."  
        fi  
    fi  
else  
    echo "Veuillez entrer un nom d'utilisateur !"  
fi
```

➤ Exercice 3

Script listant les répertoires dont l'utilisateur possède le droit d'écriture. On améliore ensuite le script pour que la recherche s'applique aussi aux sous-répertoires.

```
for a in *  
do  
    if [ -d $a ]  
    then  
        if [ -w $a ]  
        then echo "`pwd`/$a"  
        fi  
        if [ -x $a ]  
        then ( cd $a; listwrite; )  
        fi  
    fi  
done
```

➤ Exercice 4

Script recherchant dans le répertoire courant un fichier donne en paramètre et le supprimant. Modification de ce script pour que la recherche et la suppression s'applique également aux sous-répertoires.

```
if [ -n "$1" ]
then
    if [ -f "$1" -a -w . ]
    then
        echo "Effacement de `pwd`/$1"
        rm $1
    fi
    for a in *
    do
        if [ -d $a -a -x $a ]
        then ( cd $a; suppr $1; )
        fi
    done
else
    echo "Il me faut un nom de fichier !"
fi
```

➤ Exercice 5

Script « del » prenant en paramètre le nom d'un fichier, nomfic par exemple, puis le renommant en .nomfic.a

```
if [ -n "$1" ]
then
    if [ -f "$1" -a -w "$1" ]
    then
        echo "Le fichier `pwd`/$1 a ete mis a la corbeille."
        mv "./$1" ".$1.a"
    else
        echo "Fichier non trouve ou protege."
    fi
else
    echo "Veuillez specifier un nom de fichier qui vous encombre."
fi
echo "Tapez net pour vider la corbeille !"
```

Script « net » supprimant tous les fichiers créés par « del » dans le répertoire courant et ses sous-répertoires.

```
find ~ -name ".*.a" -exec rm {} \;
echo "Corbeille videe !"
```

Question subsidiaire... Script « undel » qui restaure un fichier de la corbeille :

```
if [ -n "$1" ]
then
  if [ -f ".$1.a" ]
  then
    if [ -f "$1" ]
    then
      echo "Attention, $1 existe deja dans ce repertoire !"
      echo "Renommez-le avant de restaurer l'ancienne version..."
    else
      mv ".$1.a" "$1"
      echo "Le fichier `pwd`/$1 a ete restaure !"
    fi
  else
    echo "Fichier non trouve ou definitivement perdu..."
  fi
else
  echo "Veuillez specifier un nom de fichier que vous voulez restaurer."
fi
```

Dernière petite question subsidiaire... Script « dels » (s pour state) qui donne le contenu de la corbeille :

```
echo "Contenu de la corbeille pour $USER :"
```

```
find ~ -name ".*.a" -exec echo {} \;
```