

TP UNIX 2

NICOLAS Jerome

RAVAUX Alexandre

I - Redirection des E/S - Utilisation de filtres

a. cat permet d'afficher le contenu du fichier. Le contenu du fichier est redirigé vers la commande tr grâce au |. La commande tr travaille donc sur le contenu, en l'occurrence, il convertit des minuscules aux majuscules dans l'intervalle [a-z].

La sortie de la commande est redirigée grâce à > dans le fichier "resultat".

b. Pour qu'une commande puisse être utilisée comme filtre, la commande doit pouvoir travailler sur les données en passant par les paramètres.

Mais certaines commandes ne peuvent pas être utilisées comme filtres malgré tout, donc pour en être sûr, on peut le vérifier dans le manuel.

c. Si ce n'est pas le cas, on peut utiliser de façon usuelle.

exemple : tr "[a-z]" "[A-Z]" < fichier > resultat

Il peut y avoir sur certaines commandes une différence dans le résultat. Mais elle sera minime.

Exemple :

```
20140113@tuxws01:~/Unix/TP2$ cat fichier | wc
```

```
1      5     29
```

```
20140113@tuxws01:~/Unix/TP2$ wc fichier
```

```
1  5 29 fichier
```

d. Expression équivalente : sort -n fichier

e. Par exemple : echo

Compositions de commandes:

f. `cat/etc/passwd | sort`

g. `cat /etc/passwd | grep "bash" | wc -l`

`cat /etc/passwd | grep "bash$" | wc -l`

h. `cat /etc/services | grep "tcp" | wc -l`

`cat /etc/services | grep "udp" | wc -l`

i.

Pour comprendre les questions suivantes regarder :

<http://fr.openclassrooms.com/informatique/cours/reprenez-le-contrôle-a-l-aide-de-linux/extraire-trier-et-filtrer-des-données>

j. `cat /etc/services | cut -d '#' -f3 | sort -nr | uniq`

k. `cat /etc/services | cut -d '#' -f7 | sort | uniq -c | sort -n`

II - Ecriture de scripts simples

Script simple :

a.

`monarbo.sh`

`#!/bin/bash`

`path_user=$HOME`

`mkdir -p $path_user/Unix/TP2/arbo/{src,obj}/app1`

`mkdir -p $path_user/Unix/TP2/arbo/{obj,bin}/tmp`

`mkdir -p $path_user/Unix/TP2/arbo/src/old`

`touch $path_user/Unix/TP2/arbo/src/app1/{a.c,b.c,c.c}`

`touch $path_user/Unix/TP2/arbo/obj/app1/{a.o,b.o,c.o}`

`touch $path_user/Unix/TP2/arbo/bin/app1`

b.

rendre executable : `chmod +x monarbo.sh`

executer : `./monarbo.sh`

utiliser la commande `find` pour retrouver le répertoire

c. Non car le chemin indiqué permet de créer l'arbo dans
`/users/20140113/Unix/TP2` et non dans `/users/20140113/Unix/TP2/bis`

`monarbo2.sh`

d.

```
#!/bin/bash
```

```
path_user=$HOME
```

```
mkdir -p ./arbo/{src,obj,bin}/app1
```

```
mkdir -p ./arbo/{obj,bin}/tmp
```

```
mkdir -p ./arbo/src/old
```

```
touch ./arbo/src/app1/{a.c,b.c,c.c}
```

```
touch ./arbo/obj/app1/{a.o,b.o,c.o}
```

Transmission des variables :

```
#!/bin/bash
```

```
var=3
```

```
echo "var=" $va
```

Transmission des paramètres :

i.

```
#!/bin/bash
```

```
for i in $*;do
```

```
echo "Pi=" $i
```

```
done
```

j.

il va lister tous les éléments de `/etc/*` ainsi que les éléments qui
constituent la date

k.

III - Ecriture de scripts évolués :

Script surveillance.sh :

Pas réussi. Début du programme :

```
#!/bin/bash
# Si le nombre d'arguments est inferieur a 3 ce n'est pas bon
if [ $# -lt 3 ]
then
    echo "Utilisation : surveillance <periode> <seuil> <log>"
else
    echo "Debut de la surveillance"
    # TODO : une boucle avec le sleep
    datecourante=`date`
    # TODO : filtrer la commande ps
    # listing=`ps`
    # Comptage
    cmpt=`$listing | wc -l` # affiche le nombre de lignes
fi
```

Script corbeille.sh :

A)

```
if [[ $1 == "-l" ]]; then ls corbeille
elif [[ $1 == "-v" ]]; then rm -r corbeille/*
else
    for i in $*;
    do cp $i corbeille
    rm $i
    done
fi
```

B)

```
echo "voulez vous creer une corbeille (o/n) ?"
read a
if [ "$a" == "o" ]
then
echo "entrez le nom de votre corbeille :"
read corbeille
mkdir $corbeille
else
corbeille=corbeille
fi
```

```
if [[ $1 == "-l" ]]; then ls $corbeille
elif [[ $1 == "-v" ]]; then rm -r $corbeille/*
else
for i in $*;
do cp $i $corbeille
rm $i
done
fi
```

C)

```
for i in $*;
do
b=`ls | grep $i`
if [ $i != $b ]
then
cp $i $corbeille
rm $i
else
```

```
echo "il y a deja un fichier de meme nom dans la corbeille"
fi
```

D)

A chaque fois que l'on met un fichier a la corbeille on le renomme avec son nom et son repertoire.

On compare les répertoires et s'ils sont différents l'un des deux fichiers sera renommé puis mis à la corbeille.

Script calc.sh :

```
if [ $2 == "x" ]
then
echo "le resultat est " $((($1 * $3))
elif [ $2 == "+" ]
then
echo "le resultat est " $((($1 + $3))
elif [ $2 == "-" ]
then
echo "le resultat est " $((($1 - $3))
elif [ $2 == "/" ]
then
echo "le resultat est " $((($1 / $3))
else
echo "mauvais operateur"
export $var
echo $var
fi
```