

TP4

Nom :CHARDAN
Prénom :Anaël
Groupe :C'

Compte rendu TP M 1 101 USI 1
Le 02 octobre 2013

tar -xf TP4.tgz

<1> Filtres et redirection
1.1 Redirection simples

Exercice 1:

find ~ : affiche tout ce que contient : /info-nfs/users/anachardan
find ~ > liste-fichiers : les résultats sont sur le fichier : liste-fichiers
cat -n : L'option à numéroté chaque ligne du liste-fichiers et donc chaque éléments trouvés par find.
wc : donne le nombre de ligne, de mots et d'octets d'un fichier donc de liste-fichiers
find ~ | more : combinaison de more et de find permet de lire le résultat de liste-fichiers ligne par ligne ou page par page
find ~ | wc: affiche directement le contenu du résultat de find sans passer par un autre fichier. Il faut ajouter -l pour ne voir que le nombre de ligne.
echo abcd | wc: une ligne, mot et d'octet donc une ligne, un mot et 5 octets (un de plus que le nombre de lettres tapées)
echo "Bienvenue à l'IUT" > message : Ecrit ceci dans un fichier
date >> message : Ecrit ceci à la FIN du fichier message
tr "[a-z]" "[A-Z]" < message : passe toutes les lettres en majuscule du fichier message sans le modifier

<2> Quelques filtres en détail
<2.1> grep

Exercice n2:

grep BORDEAUX insee.csv à passer en rouge chaque BORDEAUX du fichier insee.csv
grep -n BORDEAUX insee.csv : Préfixer chaque ligne de sortie par le numéro de la ligne dans le fichier. La numérotation commence à la ligne 1.
grep ^BORDEAUX insee.csv : ne donne que les résultats où BORDEAUX est en début de ligne
grep 'ST PIERRE' insee.csv: Les ' ' permettent de prendre en compte une recherche contenant un espace
grep '^ST PIERRE' insee.csv: combine l'espace et le fait de ne prendre en compte que les résultats où ST PIERRE est en début de ligne.
grep '33210' insee.csv: met en évidence les lignes où il y a 33210
grep '33210\$' insee.csv: met en évidence les lignes où 33210 est écrit en fin de ligne.
grep '33210;' insee.csv: met en évident que ceux qui sont entre ";"
^ affiche les résultats en début de ligne et \$ les résultats en fin de ligne

Exercice 3:

\$1 est un paramètre

```
#!/bin/bash
echo Commune choisie = $1
grep "'^$1;" insee.csv
```

Il faut mettre "ST MARTIN" entre guillemet
ou dans le script:

```
#!/bin/bash
echo Commune choisie = $*
grep "^$*;" insee.csv
```

-i modifie la casse et donc il y a le choix.

Exercice 4:

```
#!/bin/bash
echo Département choisie = $1
grep -i ";$1...;" insee.csv
```

<2.2> La commande cut

Exercice 5:

```
cut -d ';' -f 3 insee.csv : affiche le troisième champ délimiter par un
point virgule
cut -d ';' -f 1,3 insee.csv: affiche le premier et le troisième champ par
un ponint virgule
cut -d ';' -f 1-3 insee.csv, affiche du premier au troisième champ séparé
par des points vrgule
cut -c 1-10 insee.csv : affiche les 10 premiers caractères
```

```
#!/bin/bash
echo Département choisie = $1
grep -i ";$1...;" insee.csv | cut -d ';' -f 1,3
```

<2.3> head et tail

Exercice 6

```
head -n 20 insee.csv : Les 20 premières lignes sont affichées
head -c 20 insee.csv : Les 20 premiers octets sont affichées
tail -n 20 insee.csv : Les 20 dernières lignes sont affichées
tail -c 20 insee.csv : Les 20 dernières octets sont affichées
find ~ | head -15 : Les 15 premières lignes du find
```

Exercice 7

```
insee.csv contient 38949 lignes et nous voulons lire celles entre 100 et
120
head -n 120 insee.csv | tail -n 20
```

<2.4> La commande find

```
find ~/Bibliotheque -name '*.cc'
```

Exercice 8

```
find /var/log '*.log' | wc : Il y'en à 84.
```

xargs applique une autre commande à chacun des noms qui lui sont transmis.

Exercice 9