

**NOLLE**

**Damien**

**L3 Informatique**

**SR – Devoir 1 :**

<b><u>Note :</u></b>	<b><u>Observation :</u></b>
<b>/20</b>	

Exercice 1)

Q1 :

1. Pour créer un répertoire, il faut utiliser la commande mkdir. On veut en créer un dans notre répertoire d'accueil. Pour cela : soit on utilise la variable d'environnement \$HOME, soit on utilise le tilde « ~ ». L'objectif est de venir créer, dans notre répertoire d'accueil, le répertoire Devoir1 et dans celui-ci, le répertoire Exercice1. Ce qui nous donnera les deux commandes suivantes :

mkdir ~/Devoir1

mkdir ~/Devoir1/Exercice1

Pour éviter de les exécuter une à une, on peut venir réunir les deux commandes sur une seule ligne avec « && » (permet d'exécuter la première commande et si le code de retour de celle-ci est égal à 0, alors la seconde commande est exécutée) ou « ; » (permet d'exécuter les commandes de manière séquentielle, la première est exécutée et la fin de celle-ci est attendu avant d'exécuter la seconde).

Ce qui nous donne : mkdir ~/Devoir1 && mkdir ~/Devoir1/Exercice1

```
user@cours:~$ mkdir ~/Devoir1 && mkdir ~/Devoir1/Exercice1
user@cours:~$ ls -R
.:
Devoir1
./Devoir1:
Exercice1
./Devoir1/Exercice1:
```

2. Pour afficher les 3 premiers caractères de chaque ligne du fichier /etc/passwd, il faut dans un premier temps utiliser la commande cat pour afficher le contenu de ce dernier. Puis on va utiliser un pipe « | » qui va permettre de rediriger la sortie d'une commande vers l'entrée

d'une autre commande. La deuxième commande sera cut avec l'option -c qui permet de découper chaque ligne par caractères, en précisant l'intervalle 1-3.

Ce qui nous donne : `cat /etc/passwd | cut -c1-3`  
`user@cours:~$ cat /etc/passwd | cut -c1-3_`

Extrait du résultat :

```
man
lp:
mai
new
uuc
pro
www
bac
```

3. Pour afficher les 10 premiers noms de logins du fichier /etc/passwd, il faut, comme précédemment, afficher le contenu du fichier avec la commande cat. On utilise une pipe afin de rediriger les lignes du fichier précédent vers la commande head -n, qui va nous permettre de récupérer les n premières lignes (ici, n sera égal à 10). On redirige, avec une pipe, ces 10 premières lignes vers la commande cut -f1 -d":", cela va nous permettre de récupérer le premier champ de la ligne délimité par un « : ».

Ce qui nous donne : `cat /etc/passwd | head -n 10 | cut -f1 -d":"`  
`user@cours:~$ cat /etc/passwd | head -n 10 | cut -f1 -d":"`  
root  
daemon  
bin  
sys  
sync  
games  
man  
lp  
mail  
news

4. Chaque ligne du fichier /etc/passwd représente un utilisateur, il suffit d'utiliser la commande wc -l pour compter le nombre de lignes du contenu du fichier affiché avec la commande cat, les deux commandes reliées par une pipe. Dans l'objectif d'afficher le nombre d'utilisateurs.

Ce qui nous donne : `cat /etc/passwd | wc -l`  
`user@cours:~$ cat /etc/passwd | wc -l`  
42

5. Pour obtenir la liste des utilisateur (nom prénom) triée par ordre alphabétique dans le fichier « utilisateur » du répertoire ~/Devoir1/Exercice1. Il faut dans un premier temps afficher le contenu du fichier /etc/passwd, puis récupérer le 5<sup>ème</sup> champ et trier par ordre alphabétique avec le résultat obtenu. Enfin, on redirige la sortie standard de la dernière commande vers ~/Devoir1/Exercice1/utilisateur.

Ce qui nous donne : `cat /etc/passwd | cut -f5 -d":" | sort > ~/Devoir1/Exercice1/utilisateur`

Présence du fichier créé :

```
user@cours:~$ cat /etc/passwd | cut -f5 -d":" | sort > ~/Devoir1/Exercice1/utilisateur
user@cours:~$ cd Devoir1/Exercice1/
user@cours:~/Devoir1/Exercice1$ ls
utilisateur
```

Contenu du fichier créé (affiché avec la commande cat) :

```
backup
bin
daemon
games
Gnats Bug-Reporting System (admin)
ircd
lp
mail
Mailing List Manager
man
news
nobody
proxy
root
sync
sys
systemd Core Dumper
systemd Network Management,,,
systemd Resolver,,,
systemd Time Synchronization,,,
User,,,
uucp
www-data
```

*On peut voir que certaines lignes sont vides, cela s'explique par le fait que rien n'a été saisi dans ce champ au moment de la création de l'utilisateur.*

Q2 :

1. Pour créer un fichier, il existe différentes méthodes utilisant cat, vi, nano ou encore touch. La méthode utilisée ici est celle avec cat. Pour cela, il suffit de rediriger la sortie standard cat vers le fichier que l'on souhaite créer. On vient ensuite ajouter la commande suivante sur la même ligne que cat : << FIN. Cela permet lorsqu'on saisit le mot préciser après << (ici FIN), d'arrêter la saisi. Ce mot ne sera pas présent dans le fichier.

Ce qui nous donne : cat > ~/Devoir1/Exercice1/text1.tex << FIN

```
user@cours:~$ cat > ~/Devoir1/Exercice1/text1.tex << FIN
> C'est un petit fichier
> qui contient le mot soleil
> en plusieurs exemplaires
> pour effectuer des recherches
> sur le mot soleil
> FIN
```

Présence du fichier créé :

```
user@cours:~$ cd Devoir1/Exercice1/
user@cours:~/Devoir1/Exercice1$ ls
text1.tex  utilisateur
```

Affichage de son contenu :

```
user@cours:~/Devoir1/Exercice1$ cat text1.tex
C'est un petit fichier
qui contient le mot soleil
en plusieurs exemplaires
pour effectuer des recherches
sur le mot soleil
```

2. Pour copier le fichier « text1.tex » précédemment créé dans deux fichier « 1-text.tex » et « 2-text.tex », il faut utiliser la commande cp. Il faut en réaliser deux qui seront réunis sur la même ligne à l'aide de « ; ».

Ce qui donne : cp ~/Devoir1/Exercice1/text1.tex ~/Devoir1/Exercice1/1-text.tex; cp  
~/Devoir1/Exercice1/text1.tex ~/Devoir1/Exercice1/2-text.tex

```
user@cours:~$ cp ~/Devoir1/Exercice1/text1.tex ~/Devoir1/Exercice1/1-text.tex; cp ~/Devoir1/Exercice1/text1.tex ~/Devoir1/Exercice1/2-text.tex
```

Présence des deux fichiers :

```
user@cours:~$ cd Devoir1/Exercice1/
user@cours:~/Devoir1/Exercice1$ ls
1-text.tex 2-text.tex text1.tex utilisateur
```

Contenu des deux fichiers :

```
user@cours:~/Devoir1/Exercice1$ echo "Contenu de 1-text.tex :"; cat 1-text.tex; echo; echo "Contenu de 2-text.tex :"; cat 2-text.tex
Contenu de 1-text.tex :
C'est un petit fichier
qui contient le mot soleil
en plusieurs exemplaires
pour effectuer des recherches
sur le mot soleil

Contenu de 2-text.tex :
C'est un petit fichier
qui contient le mot soleil
en plusieurs exemplaires
pour effectuer des recherches
sur le mot soleil
```

3. Pour créer un fichier vide, on va utiliser la commande touch. Il suffit de lui préciser le chemin où se trouveront les futurs fichiers.

Ce qui donne : touch ~/Devoir1/test1.core; touch ~/Devoir1/Exercice1/test2.core

```
user@cours:~$ touch ~/Devoir1/test1.core; touch ~/Devoir1/Exercice1/test2.core
```

Présence du fichier « test1.core » dans le répertoire Devoir1 :

```
user@cours:~$ cd Devoir1/  
user@cours:~/Devoir1$ ls  
Exercice1 test1.core
```

Contenu du fichier « test1.core » :

```
user@cours:~/Devoir1$ cat test1.core  
user@cours:~/Devoir1$
```

*On voit que le fichier est bien vide.*

Présence du fichier « test2.core » dans le répertoire Devoir1/Exercice1 :

```
user@cours:~$ cd Devoir1/Exercice1/  
user@cours:~/Devoir1/Exercice1$ ls  
1-text.tex 2-text.tex test2.core text1.tex utilisateur
```

Contenu du fichier « test2.core » :

```
user@cours:~/Devoir1/Exercice1$ cat test2.core  
user@cours:~/Devoir1/Exercice1$
```

*On voit que le fichier est bien vide.*

4. Pour rechercher une chaîne dans un fichier, il faut utiliser la commande grep. Cette dernière permet, via l'utilisation d'une expression régulière (regular expression ou RegEx), de ressortir les lignes contenant les occurrences recherchées.

Ce qui nous donne : grep "soleil" ~/Devoir1/Exercice1/text1.tex

```
user@cours:~$ grep "soleil" ~/Devoir1/Exercice1/text1.tex  
qui contient le mot soleil  
sur le mot soleil
```

5. Pour rechercher tous les fichiers du répertoire courant (.) ayant dans leurs noms l'extension « .tex » (-name), il faut utiliser la commande find. Par la suite, il faut que cette dernière puisse exécuter (-exec) la commande grep afin de rechercher la chaîne « soleil » dans tous les fichiers trouvés.

Ce qui nous donne : find . -name \*.tex -print -exec grep "soleil" {} \;

```
user@cours:~$ find . -name *.tex -print -exec grep "soleil" {} \;  
./Devoir1/Exercice1/2-text.tex  
qui contient le mot soleil  
sur le mot soleil  
./Devoir1/Exercice1/text1.tex  
qui contient le mot soleil  
sur le mot soleil  
./Devoir1/Exercice1/1-text.tex  
qui contient le mot soleil  
sur le mot soleil
```

6. Pour recherche dans les fichiers du répertoire de l'utilisateur courant (~) les fichiers ayant dans leurs noms l'extension « .core » (-name), il faut utiliser la commande find.

```
Ce qui nous donne : find ~ -name *.core -print
user@cours:~$ find ~ -name *.core -print
/home/user/Devoir1/test1.core
/home/user/Devoir1/Exercice1/test2.core
```

7. Pour supprimer tous les fichiers ayant l'extension « .core » dans le répertoire de l'utilisateur courant, il suffit de reprendre la commande précédente et de rajouter « -exec rm {} \; » en supprimant « -print ».

```
Ce qui nous donne : find ~ -name *.core -exec rm {} \;
user@cours:~$ find ~ -name *.core -exec rm {} \;
```

Preuve de la suppression du fichier /Devoir1/test1.core :

```
user@cours:~$ cd Devoir1/
user@cours:~/Devoir1$ ls
Exercice1
```

*On remarque que le fichier « test1.core » n'est plus présent dans le répertoire Devoir1, il a donc bien été supprimé.*

Preuve de la suppression du fichier /Devoir1/Exercice1/test2.core :

```
user@cours:~$ cd Devoir1/Exercice1/
user@cours:~/Devoir1/Exercice1$ ls
1-text.tex 2-text.tex text1.tex utilisateur
```

*On remarque que le fichier « test2.core » n'est plus présent dans le répertoire Devoir1/Exercice1, il a donc bien été supprimé.*

8. Pour afficher les fichiers du répertoire ~/Devoir1/Exercice1 dont le nom ne commence pas par une lettre, il faut utiliser la commande find ayant en option -name : [^a-zA-Z]\*.

```
Ce qui nous donne : find ~/Devoir1/Exercice1 -name [^a-zA-Z]* -print
user@cours:~$ find ~/Devoir1/Exercice1/ -name [^a-zA-Z]* -print
/home/user/Devoir1/Exercice1/2-text.tex
/home/user/Devoir1/Exercice1/1-text.tex
```

9. Pour afficher le contenu des fichiers du répertoire ~/Devoir1/Exercice1 dont le nom ne commence pas par une lettre, il faut reprendre la commande précédente et ajouter « -exec cat {} \; ».

```
Ce qui nous donne : find ~/Devoir1/Exercice1 -name [^a-zA-Z]* -print -exec cat {} \;
user@cours:~$ find ~/Devoir1/Exercice1/ -name [^a-zA-Z]* -print -exec cat {} \;
/home/user/Devoir1/Exercice1/2-text.tex
C'est un petit fichier
qui contient le mot soleil
en plusieurs exemplaires
pour effectuer des recherches
sur le mot soleil
/home/user/Devoir1/Exercice1/1-text.tex
C'est un petit fichier
qui contient le mot soleil
en plusieurs exemplaires
pour effectuer des recherches
sur le mot soleil
```

## Exercice 2)

Q1 : Tentative de suppression du fichier /etc/passwd avec la commande rm :

```
user@cours:~$ rm /etc/passwd
rm : supprimer '/etc/passwd' qui est protégé en écriture et est du type « fichier » ? o
rm: impossible de supprimer '/etc/passwd': Permission non accordée
```

Affichage du fichier /etc/passwd et de ces informations avec la commande ls -l :

```
user@cours:~$ ls -l /etc/passwd
-rw-r--r-- 1 root root 2400 oct. 11 2020 /etc/passwd
```

Affichage des informations sur le dossier /etc avec la commande ls -l :

```
user@cours:~/Devoir1$ ls -l / | grep "etc"
drwxr-xr-x 70 root root 4096 déc. 3 11:42 etc
```

Affichage des informations de l'utilisateur root présents dans /etc/passwd :

```
user@cours:~$ grep "root" /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

Affichage des informations du groupe principal de l'utilisateur root présents dans /etc/group :

```
user@cours:~$ grep "root" /etc/group
root:x:0:
```

On remarque que le fichier /etc/passwd ne s'est pas supprimé car la permission n'est pas accordée. En regardant les droits accordés sur ce fichier, on se rend compte que :

- L'utilisateur propriétaire a le droit de lecture (r) et d'écriture (w) sur ce fichier.
- Le groupe et les autres ont seulement un droit de lecture (r) sur ce fichier.

Et en regardant les droits accordés sur le répertoire /etc, on se rend compte que :

- L'utilisateur propriétaire a le droit de lire et d'afficher avec la commande ls le contenu du répertoire (r), de modifier le contenu (suppression et ajout de fichiers) du répertoire (w) et d'accéder aux fichiers contenus dans ce répertoire (x).
- Le groupe et les autres ont seulement le droit de lire et d'afficher le contenu du répertoire avec la commande ls (r) et d'accéder aux fichiers contenus dans ce répertoire (x).

Etant donné que nous sommes l'utilisateur « user » (donc pas propriétaire du fichier /etc/passwd ni du répertoire /etc) et que nous n'appartenons pas au groupe principal de l'utilisateur propriétaire, les droits qui nous sont accordés sont ceux des autres (donc de lecture seulement sur le fichier et de lecture, d'affichage avec la commande ls ainsi que d'accès au contenu du répertoire /etc).

Les droits sur le fichier expliquent le premier message nous demandant si on souhaite supprimer un fichier protégé en écriture, car nous pouvons consulter le contenu du fichier mais nous ne pouvons pas le modifier.

Les droits sur le répertoire expliquent pourquoi nous ne pouvons pas supprimer le fichier, car nous n'avons ni le droit d'ajout ni le droit de suppression de fichiers (w) sur le répertoire /etc.

Q2 :

1. Pour qu'un fichier texte soit lisible mais non modifiable par tout le monde (y compris pour l'utilisateur propriétaire), il faut positionner les droits comme suivant : r--r--r--.

Pour se faire, on peut utiliser la commande chmod en écriture rwx de la manière suivant :

```
chmod a=r unFichier
```

Ou en écriture octal :

u : 100

g : 100

o : 100

$$100_{(2)} = 2^2 * 1 + 2^1 * 0 + 2^0 * 0 = 4 * 1 + 0 + 0 = 4$$

Donc :

```
chmod 444 unFichier
```

2. Pour qu'un répertoire nommé « topsecret » n'ai son contenu modifiable et visible que par son utilisateur propriétaire, il faut positionner les droits comme suivant : rwx----

Avec la commande chmod en écriture rwx :

```
chmod u=rwx g=o topsecret
```

En écriture octal :

u : 111

g : 000

o : 000

$$111_{(2)} = 2^2 * 1 + 2^1 * 1 + 2^0 * 1 = 4 * 1 + 2 * 1 + 1 * 1 = 7$$

$$000_{(2)} = 0$$



Donc :

chmod 700 topsecret

3. Pour qu'un utilisateur quelconque ne puisse pas lister les fichiers du répertoire « petitsecret » mais puisse les lire, il faut positionner les droits comme suivant : rwx--x--x.

Avec la commande chmod en écriture rwx :

chmod u=rwx g=x o=g petitsecret

En écriture octal :

u : 111

g : 001

o : 001

$$111_{(2)} = 2^2 * 1 + 2^1 * 1 + 2^0 * 1 = 4 * 1 + 2 * 1 + 1 * 1 = 7$$

$$001_{(2)} = 2^2 * 0 + 2^1 * 0 + 2^0 * 1 = 0 + 0 + 1 * 1 = 1$$

Donc :

chmod 711 petitsecret

Si un utilisateur quelconque appartient au groupe de l'utilisateur propriétaire, il aura les mêmes droits que les autres. Il pourra accéder aux fichiers présents dans le répertoire petitsecret mais il ne pourra pas lister le contenu de ce dernier.

Q3 :

1. Pour que l'utilisateur *dupond* puisse recopier le fichier *f* situé dans son répertoire d'accueil, dans le répertoire d'accueil d'un utilisateur *antoine*. Il y a plusieurs conditions à remplir :
  - Soit l'utilisateur *dupond*, appartient au groupe principal de l'utilisateur *antoine*. Dans ce cas-là, ce dernier positionne le droit de modification du contenu de son répertoire d'accueil (w) au groupe.
  - Soit l'utilisateur *dupond* n'appartient pas au groupe principal de *antoine* mais il appartient à un groupe auquel ce dernier l'est aussi. Dans ce cas-là, ce dernier modifie le groupe d'appartenance de son répertoire d'accueil avec `chown` et positionne le droit de modification du contenu de son répertoire d'accueil (w) à ce nouveau groupe.
  - Soit *dupond* n'appartient à aucun groupe où *antoine* est présent. Dans ce cas-là, ce dernier positionne le droit de modification du contenu de son répertoire d'accueil (w) aux autres.
2. Pour que l'utilisateur *dupond* puisse effacer un fichier *fich1* situé dans le répertoire d'accueil d'un utilisateur *antoine*. Il y a plusieurs conditions à remplir :
  - Soit l'utilisateur *dupond*, appartient au groupe principal de l'utilisateur *antoine*. Dans ce cas-là, ce dernier positionne le droit de modification du contenu de son répertoire d'accueil (w) au groupe.
  - Soit l'utilisateur *dupond* n'appartient pas au groupe principal de *antoine* mais il appartient à un groupe auquel ce dernier l'est aussi. Dans ce cas-là, ce dernier

modifie le groupe d'appartenance de son répertoire d'accueil avec `chown` et positionne le droit de modification du contenu de son répertoire d'accueil (`w`) à ce nouveau groupe.

- Soit *dupond* n'appartient à aucun groupe où *antoire* est présent. Dans ce cas-là, ce dernier positionne le droit de modification du contenu de son répertoire d'accueil (`w`) aux autres.

### Exercice 3)

1. Création du sous-répertoire « Exercice3 » du répertoire `~/Devoir1` avec la commande `mkdir` :

```
user@cours:~$ mkdir ~/Devoir1/Exercice3
```

Présence du répertoire « Exercice3 » dans le répertoire « Devoir1 » :

```
user@cours:~$ ls -l ~/Devoir1/
total 8
drwxr-xr-x 2 user user 4096 déc.  3 16:22 Exercice1
drwxr-xr-x 2 user user 4096 déc.  3 19:07 Exercice3
```

2. La commande permettant d'afficher « Bonjour » à l'écran et en même temps de créer un fichier « `salutation.txt` » dans le répertoire « Exercice3 », précédemment créé, dont le contenu est le même message, est la suivante :

```
user@cours:~$ echo Bonjour | tee ~/Devoir1/Exercice3/salutation.txt
```

Preuve de l'affichage du message « Bonjour » à l'écran :

```
user@cours:~$ echo Bonjour | tee ~/Devoir1/Exercice3/salutation.txt
Bonjour
```

Preuve de la création du fichier « `salutation.txt` » dans le répertoire `~/Devoir1/Exercice3` :

```
user@cours:~$ ls -l ~/Devoir1/Exercice3
total 4
-rw-r--r-- 1 user user 8 déc.  3 19:15 salutation.txt
```

Et contenant le même message que celui affiché :

```
user@cours:~$ cat ~/Devoir1/Exercice3/salutation.txt
Bonjour
```

3. Création du script shell « lectEcriture » qui permet de passer les caractères en minuscule, en majuscule :

- Création du fichier du script avec nano :

```
user@cours:~$ nano ~/Devoir1/Exercice3/lectEcriture_
```

- Code du script :

```
GNU nano 3.2 /home/us
#!/bin/bash
while read ch
do
    echo $ch | tr 'a-z' 'A-Z'
done
```

*La commande read permet de lire sur l'entrée standard et d'affecter la valeur à une variable, ici nommé ch. La boucle while permet ici de lire, si sur l'entrée standard il y a plusieurs lignes, chaque ligne (délimité par un retour à la ligne). On appelle ensuite la commande echo avec cette dernière qui permet, en temps normal, d'afficher la valeur de la variable sur la sortie standard. Sauf qu'une pipe redirige la sortie standard de la commande echo vers l'entrée standard de la commande tr. Cette dernière permet de remplacer chaque caractère dans l'intervalle [a-z] (les caractères en minuscules) en son équivalent dans l'intervalle [A-Z] (en majuscules).*

- Ajout du droit d'exécution sur le shell script « lectEcriture » pour l'utilisateur propriétaire (nous) :

```
user@cours:~$ chmod u+x ~/Devoir1/Exercice3/lectEcriture_
```

- Ajout du chemin où se trouve « lectEcriture » dans la variable d'environnement PATH de manière à pouvoir exécuter le shell script en l'appelant uniquement par le nom du fichier :

```
user@cours:~$ PATH=$PATH":~/Devoir1/Exercice3" _
```

4. Modification du script précédent, afin qu'il écrive le nombre de caractères traités dans un fichier :

```
GNU nano 3.2 /home/user/Devoir1/Exercice3/lectEcriture

#!/bin/bash

cpt_car_traites=0

while read ch
do
    s_ch=$((echo $ch | wc -c) - 1)
    i=1

    while [ $i -le $s_ch ]
    do
        if [ $(echo $ch | cut -c $i | grep -Ec "[a-z]") -eq 1 ]
        then
            cpt_car_traites=$((cpt_car_traites+1))
        fi
        i=$((i+1))
    done

    echo $ch | tr 'a-z' 'A-Z'
done

echo $cpt_car_traites > ~/Devoir1/Exercice3/fich_nb_car_traites
```

La variable `s_ch` contient le nombre de caractères réel de la chaîne (-1 car le caractère de retour à la ligne est compté avec la commande `wc -c`). La variable `i` est un compteur qui va nous permettre de passer en revue chaque caractère de la chaîne, elle est incrémentée à chaque itération dans la boucle `while`. Ces deux variables vont former la condition permettant de sortir de cette boucle (tant que la valeur de `i` est inférieure ou égale à celle de la taille, on reste dans la boucle). A chaque itération, on va venir regarder si chaque caractère de `ch` correspond à une minuscule (`grep -Ec` nous permet de compter le nombre de ligne ayant un ou plusieurs caractères correspondant à la RegEx. Ici, vu que l'on teste cette dernière pour un seul caractère : si cette commande retourne 1, alors le caractère est une minuscule, 0 sinon). Si le caractère testé est une minuscule, on incrémente alors la variable `cpt_car_traites`. Cette dernière correspond au nombre de caractères traités. On finit par afficher la valeur de `ch` en majuscule sur la sortie standard et on utilise la commande `echo` en redirigeant sa sortie standard dans un fichier pour y écrire la valeur de `cpt_car_traites`.

5. Pour recopier le contenu passé en majuscule du fichier ~/Devoir1/Exercice1/text1.tex dans un fichier text1-maj.tex (qui sera sauvegardé dans le répertoire ~/Devoir1/Exercice3) et l'afficher sur l'écran tout en affichant le nombre de caractères traités, il faut utiliser la commande suivante : `cat ~/Devoir1/Exercice1/text1.tex | lectEcriture | tee ~/Devoir1/Exercice3/text1-maj.tex; cat ~/Devoir1/Exercice3/fich_nb_car_traites`

```
user@cours:~$ cat ~/Devoir1/Exercice1/text1.tex | lectEcriture | tee ~/Devoir1/Exercice3/text1-maj.t
ex; cat ~/Devoir1/Exercice3/fich_nb_car_traites
C'EST UN PETIT FICHIER
QUI CONTIENT LE MOT SOLEIL
EN PLUSIEURS EXEMPLAIRES
POUR EFFECTUER DES RECHERCHES
SUR LE MOT SOLEIL
101
```

Contenu du fichier ~/Devoir1/Exercice3/text1-maj.tex :

```
user@cours:~$ cat ~/Devoir1/Exercice3/text1-maj.tex
C'EST UN PETIT FICHIER
QUI CONTIENT LE MOT SOLEIL
EN PLUSIEURS EXEMPLAIRES
POUR EFFECTUER DES RECHERCHES
SUR LE MOT SOLEIL
```

#### Exercice 4)

Q1 : Création du sous-répertoire « Exercice4 » dans le répertoire « Devoir1 » avec la commande `mkdir` :

```
user@cours:~$ mkdir ~/Devoir1/Exercice4
user@cours:~$ ls ~/Devoir1
Exercice1 Exercice3 Exercice4
```

Q2 : Création du script shell « nomGroupe » dans le répertoire ~/Devoir1/Exercice4 avec la commande `nano` :

```
user@cours:~$ nano ~/Devoir1/Exercice4/nomGroupe_
```

Code du script shell « nomGroupe » :

```
#!/bin/bash

if [ $# -ne 1 ]
then
    echo "Usage de la commande : nomGroupe <Chemin vers le répertoire>"
    exit 1
elif [ ! -d $1 ]
then
    echo "Le chemin ou le nom saisi ne correspond à aucun répertoire existant !"
    exit 2
fi

find $1 -maxdepth 1 -type f -print | while read ligne
do
    echo -e "Fichier : \"$ligne\"\nNom de l'utilisateur propriétaire : \"$(ls -l $ligne | cut -f3 -d" " | tr '[A-Z]' '[a-z]')"\nGroupe d'appartenance du fichier : \"$(ls -l $ligne | cut -f4 -d" " | tr '[a-z]' '[A-Z]')"\n"
done
```

Ajout du droit d'exécution sur le script shell précédent pour l'utilisateur propriétaire (nous) :

```
user@cours:~$ chmod u+x ~/Devoir1/Exercice4/nomGroupe
```

Ajout du chemin où se trouve le script shell précédent dans la variable d'environnement PATH, afin de pouvoir l'exécuter en utilisant le nom du fichier uniquement :

```
user@cours:~$ PATH=$PATH":~/Devoir1/Exercice4"
```

### **Résultats :**

- Avec un chemin correct :

```
user@cours:~$ nomGroupe ~/Devoir1/Exercice1
Fichier : /home/user/Devoir1/Exercice1/utilisateur
Nom de l'utilisateur propriétaire : user
Groupe d'appartenance du fichier : USER

Fichier : /home/user/Devoir1/Exercice1/2-text.tex
Nom de l'utilisateur propriétaire : user
Groupe d'appartenance du fichier : USER

Fichier : /home/user/Devoir1/Exercice1/text1.tex
Nom de l'utilisateur propriétaire : user
Groupe d'appartenance du fichier : USER

Fichier : /home/user/Devoir1/Exercice1/1-text.tex
Nom de l'utilisateur propriétaire : user
Groupe d'appartenance du fichier : USER
```

- Avec un chemin incorrect :

```
user@cours:~$ nomGroupe ~/Devoir1/Exercice6
Le chemin ou le nom saisi ne correspond à aucun répertoire existant !
```

- Sans paramètre :

```
user@cours:~$ nomGroupe
Usage de la commande : nomGroupe <Chemin vers le répertoire>
```

Q3 : Création d'un fichier dans le répertoire ~/Devoir1/Exercice4, contenant un exemple de type carnet d'adresse avec la commande cat qui nous permettra de tester le futur script gestionCarnet, nommé « carnet-exemple » :

```
user@cours:~$ cat > ~/Devoir1/Exercice4/carnet-exemple << FIN
> Durand;Emilie;0381818585
> Terieur;Alex;0478858689
> Dupond;Albert;04961868957
> Dupont;Emilie;02971457895
> Dupond;Albertine;0131986258
> Bouvier;Jacques;0381698759
> Dupond;Agnès;0687598614
> Dumont;Patrick;04661645987
> Houtand;Laurent;0658769458
> FIN
```

Création du script shell « gestionCarnet » dans le répertoire ~/Devoir1/Exercice4 avec nano :

```
user@cours:~$ nano ~/Devoir1/Exercice4/gestionCarnet
```

Code du script précédemment créer :

```
GNU nano 3.2 /home/user/Devoir1/Exercice4/gestionCarnet
#!/bin/bash

if [ $# -ne 1 ]
then
    echo "Usage de la commande : gestionCarnet <Chemin vers le fichier carnet>"
    exit 1
elif [ ! -f $1 ]
then
    echo "Le chemin ou le nom saisi ne correspond pas à un fichier existant !"
    exit 2
fi

reset

while true
do
    echo -e "-----MENU-----\n
    Bienvenue dans le gestionnaire des carnets !\n
    \nVeuillez choisir l'opération en saisissant\n
    le numéro qui précède le nom de l'opération.\n
    \n1. Afficher le carnet d'adresse trié par\n
    ordre alphabétique de noms.\n2. \n
    Afficher le nombre de personnes du carnet\n
    d'adresse.\n
    3. Recherche les lignes correspondant au nom\n
    \nsaisi.\n
    4. Recherche les lignes ne correspondant pas\n
    \nau nom saisi.\n
    5. Recherche le numéro de téléphone\n
    correspondant au nom saisi.\n"

    printf "%s" "--> Veuillez saisir votre choix : "
```

```
GNU nano 3.2 /home/user/Devoir1/Exercice4/gestionCarnet

read choix
case $choix in
1) cat $1 | tr ";" " " | sort -k1 | tr " " ";";;
2) echo "Il y a "$(cat $1 | wc -l)" personne(s) dans ce carnet.;;
3) printf "%s" "--> Veuillez saisir un nom : "
   read nom

   cat $1 | while read ligne
   do
       if [ $(echo $ligne | cut -f1 -d";") = $nom ]
       then
           echo $ligne
       fi
   done;;
4) printf "%s" "--> Veuillez saisir un nom : "
   read nom

   cat $1 | while read ligne
   do
       if [ $(echo $ligne | cut -f1 -d";") != $nom ]
       then
           echo $ligne
       fi
   done;;
5) printf "%s" "--> Veuillez saisir un nom : "
   read nom
   trouve=false

   cat $1 | while read ligne && [ $trouve = false ]
   do
       if [ $(echo $ligne | cut -f1 -d";") = $nom ]
       then
           echo $(echo $ligne | cut -f3 -d";")
           trouve=true
       fi
   done;;
*) echo "Votre choix est incorrecte !";;
esac

echo -e "\n"
done
_
```

Ajout du droit d'exécution sur le script shell précédent pour l'utilisateur propriétaire (nous) :

```
user@cours:~$ chmod u+x ~/Devoir1/Exercice4/gestionCarnet
```

Pas besoins d'ajouter le chemin où se trouve le script shell dans la variable d'environnement PATH, car cela a déjà été fait à la question précédente.



### Résultats :

- Test du premier choix :

```
-----MENU-----  
Bienvenue dans le gestionnaire des carnets !  
  
Veuillez choisir l'opération en saisissant  
le numéro qui précède le nom de l'opération.  
  
1. Afficher le carnet d'adresse trié par  
ordre alphabétique de noms.  
2. Afficher le nombre de personnes du carnet  
d'adresse.  
3. Recherche les lignes correspondant au nom  
saisi.  
4. Recherche les lignes ne correspondant pas  
au nom saisi.  
5. Recherche le numéro de téléphone  
correspondant au nom saisi.  
  
--> Veuillez saisir votre choix : 1_
```

```
Bouvier;Jacques;0381698759  
Dumont;Patrick;04661645987  
Dupond;Agnès;0687598614  
Dupond;Albert;04961868957  
Dupond;Albertine;0131986258  
Dupont;Emilie;02971457895  
Durand;Emilie;0381818585  
Houtand;Laurent;0658769458  
Terieur;Alex;0478858689
```

- Test du second choix :

```
-----MENU-----  
Bienvenue dans le gestionnaire des carnets !  
  
Veuillez choisir l'opération en saisissant  
le numéro qui précède le nom de l'opération.  
  
1. Afficher le carnet d'adresse trié par  
ordre alphabétique de noms.  
2. Afficher le nombre de personnes du carnet  
d'adresse.  
3. Recherche les lignes correspondant au nom  
saisi.  
4. Recherche les lignes ne correspondant pas  
au nom saisi.  
5. Recherche le numéro de téléphone  
correspondant au nom saisi.  
  
--> Veuillez saisir votre choix : 2
```

```
Il y a 9 personne(s) dans ce carnet.
```

- Test du troisième choix :

```
-----MENU-----
Bienvenue dans le gestionnaire des carnets !

Veuillez choisir l'opération en saisissant
le numéro qui précède le nom de l'opération.

1. Afficher le carnet d'adresse trié par
ordre alphabétique de noms.
2. Afficher le nombre de personnes du carnet
d'adresse.
3. Recherche les lignes correspondant au nom
saisi.
4. Recherche les lignes ne correspondant pas
au nom saisi.
5. Recherche le numéro de téléphone
correspondant au nom saisi.

--> Veuillez saisir votre choix : 3
--> Veuillez saisir un nom : Dupond_

Dupond;Albert;04961868957
Dupond;Albertine;0131986258
Dupond;Agnès;0687598614
```

- Test du quatrième choix :

```
-----MENU-----
Bienvenue dans le gestionnaire des carnets !

Veuillez choisir l'opération en saisissant
le numéro qui précède le nom de l'opération.

1. Afficher le carnet d'adresse trié par
ordre alphabétique de noms.
2. Afficher le nombre de personnes du carnet
d'adresse.
3. Recherche les lignes correspondant au nom
saisi.
4. Recherche les lignes ne correspondant pas
au nom saisi.
5. Recherche le numéro de téléphone
correspondant au nom saisi.

--> Veuillez saisir votre choix : 4
--> Veuillez saisir un nom : Dupond_
```

```
Durand;Emilie;0381818585
Terieur;Alex;0478858689
Dupont;Emilie;02971457895
Bouvier;Jacques;0381698759
Dumont;Patrick;04661645987
Houtand;Laurent;0658769458
```

- Test du cinquième choix :

```
-----MENU-----
Bienvenue dans le gestionnaire des carnets !

Veuillez choisir l'opération en saisissant
le numéro qui précède le nom de l'opération.

1. Afficher le carnet d'adresse trié par
ordre alphabétique de noms.
2. Afficher le nombre de personnes du carnet
d'adresse.
3. Recherche les lignes correspondant au nom
saisi.
4. Recherche les lignes ne correspondant pas
au nom saisi.
5. Recherche le numéro de téléphone
correspondant au nom saisi.

--> Veuillez saisir votre choix : 5
--> Veuillez saisir un nom : Dupond_
```

```
04961868957
```

- Test d'un choix erroné :

```
-----MENU-----
Bienvenue dans le gestionnaire des carnets !

Veuillez choisir l'opération en saisissant
le numéro qui précède le nom de l'opération.

1. Afficher le carnet d'adresse trié par
ordre alphabétique de noms.
2. Afficher le nombre de personnes du carnet
d'adresse.
3. Recherche les lignes correspondant au nom
saisi.
4. Recherche les lignes ne correspondant pas
au nom saisi.
5. Recherche le numéro de téléphone
correspondant au nom saisi.

--> Veuillez saisir votre choix : 8_
```

```
Votre choix est incorrecte !
```