

Contrôle final

EXERCICE 1

Supposons avoir l'arborescence suivante :

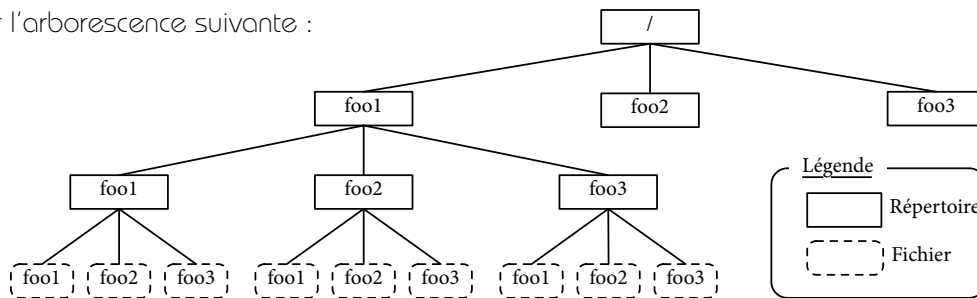


Figure 1: Exemple d'une arborescence

1. Donnez le chemin relatif du répertoire **/foo1/foo2** si le répertoire courant est :

**2 points**

Répertoire courant	Chemin relatif du répertoire <b>/foo1/foo2</b>
<b>/foo1/foo1</b>	<b>../foo2</b>
<b>/foo1/foo2</b>	<b>.</b>
<b>/foo1/foo3</b>	<b>../foo2</b>
<b>/foo1</b>	<b>foo2</b> ou <b>./foo2</b>
<b>/foo2</b>	<b>../foo1/foo2</b>
<b>/foo3</b>	<b>../foo1/foo2</b>
<b>/</b>	<b>foo1/foo2</b> ou <b>./foo1/foo2</b>

2. En utilisant une seule ligne de commande, supprimez l'ensemble des fichiers **foo1**, **foo2** et **foo3** des trois répertoire **foo1**, **foo2** et **foo3**.

Exercice 1 - Question 2 (1 point)

`rm /foo1/foo[1-3]/foo[1-3]` ou `rm /foo1/foo*/foo*`

3. En utilisant une seule ligne de commande, supprimez les trois répertoires **foo1**, **foo2** et **foo3** du répertoire **/foo1**

Exercice 1 - Question 3 (1 point)

`rmdir /foo1/foo[1-3]` ou `rm -r /foo1/foo[1-3]` ou `rmdir /foo1/foo*`

4. Sachant que le numéro d'inœud s'incrmente par un à chaque création de nouveau fichier et que le prochain numéro d'inœud libre est 1000. Donnez le résultat d'exécution du script-shell suivant :

Script-shell

```
#!/bin/bash
cd /foo1
umask 012
mkdir TP1
touch TP1/foo1
ln -s TP1/foo1 foo1
ln TP1/foo1 foo2
ls -il
```

### Exercice 1 - Question 3 (2 points)

```
total 0
1002 lrwxrwxrwx 1 etudiant1 groupe1    8 Jan 16 23:05 foo1 -> TP1/foo1
1001 -rw-rw-r-- 2 etudiant1 groupe1    0 Jan 16 23:05 foo2
1000 drwxrw-r-x 1 etudiant1 groupe1 4096 Jan 16 23:05 TP1
```

## EXERCICE 2

Écrivez un script-shell qui, étant donné un argument entier  $x$ , crée  $x$  fichiers nommés **fichier1.txt**, **fichier2.txt** et ainsi de suite jusqu'à **fichierx.txt** et leur donner tous les droits pour l'utilisateur

### Exercice 2 : (2.5 points)

```
#!/bin/bash
#Première solution
i=1
while [ $i -le $1 ]
do
    touch fichier$i.txt
    chmod u=rwx fichier$i.txt
    i=`expr $i + 1`
done
# Deuxième solution
i=1
until [ $i -gt $1 ]
do
    touch fichier$i.txt
    chmod u=rwx fichier$i.txt
    i=`expr $i + 1`
done
# Troisième solution
for i in `seq 1 $1`
do
    touch fichier$i.txt
    chmod u=rwx fichier$i.txt
done
```

## EXERCICE 3

Dans cet exercice vous voulez préparer un répertoire contenant tous vos programmes en **C** pour que votre binôme puisse le copier sur son compte. Nous supposons que tous vos programmes en **C** sont stockés dans des fichiers dont le nom se termine par **.c**, situés dans les répertoires **SE/TPx** de votre répertoire personnel où  $x$  est un numéro de **TP**.

1. Indiquez la ou les commandes permettant de créer un répertoire **Programmes** dans votre répertoire personnel et d'y copier tous vos programmes en **C**.

### Exercice 3 - Question 1 (2 points)

```
mkdir ~/Programmes
cp ~/SE/TP*/*.c ~/Programmes
```

2. En supposant que la copie ne modifie pas les droits et que :

- votre répertoire personnel a les droits **rwX-----**
- votre répertoire **Programmes** a les droits **rwXr-----**
- les fichiers contenant vos programmes **C** ont les droits **rw-r--r--**

Donnez la ou les commandes permettant de modifier les droits de vos fichiers et répertoires afin que votre binôme, qui ne fait pas partie du même groupe **Unix** que vous, puisse copier tous vos programmes **C** sur son compte.

Exercice 3 - Question 2 (1.5 point)

```
chmod o+rx ~
chmod o+rx ~/Programmes
```

3. Donnez la ou les commandes permettant de déplacer votre répertoire **Programmes** vers le répertoire **Archives** de votre répertoire personnel et de le renommer en **Programmes\_2020-2021**.

Exercice 3 - Question 3 (1 point)

```
mv ~/Programmes ~/Archives/Programmes_2020-2021
```

## EXERCICE 4

On dispose d'un fichier **eleves.txt** contenant les prénoms et noms de l'ensemble des étudiants du groupe de TD. Le fichier est de la forme suivante :

```
prenom1:nom1
prenom2:nom2
prenom3:nom3
...
```

**Remarque : Attention, pour chaque question il est possible de se servir de la réponse à l'une des questions précédentes, le nom du script associé est QX.sh où X est le numéro de la question.**

1. Écrivez un script qui lit sur l'entrée standard un texte constitué d'une seule ligne contenant un prénom et un nom séparés par le caractère ':' et affiche l'adresse mail de l'étudiant associé. Pour mémoire, l'adresse mail d'un étudiant est **prénom.nom@um5r.ac.ma**.

Exercice 4 - Question 1 (1.5 point)

```
#!/bin/bash
# script : Q1.sh
IFS=":"
read -p 'Donnez votre prénom et votre nom séparés par ":" ' prenom nom
echo "$prenom.$nom@um5r.ac.ma"
```

2. Écrivez un script qui prend en argument le nom d'un fichier au même format que le fichier **eleves.txt** donné en exemple et affiche, pour chaque ligne de ce fichier, l'adresse mail de l'étudiant associé.

Exercice 4 - Question 2 (1.5 point)

```
#!/bin/bash
# script : Q2.sh
cat $1 | while read ligne
do
    echo $ligne | ./Q1.sh
done
```

3. La commande **mail** permet d'envoyer des emails en ligne de commande. Par exemple :  
**mail -s bonjour toto@gmail.com** lit sur l'entrée standard un message et l'envoie à l'adresse mail **toto@gmail.com** (dernier argument), avec pour sujet **bonjour** (argument de l'option **-s**). Écrivez un script qui prend 3 arguments :

- le nom **a** d'un fichier au même format que le fichier **eleves.txt**

- le nom **b** d'un fichier texte
- une chaîne de caractères **c**

et pour chaque ligne du fichier **a** envoie par mail à l'adresse mail de l'étudiant associé le message contenu dans **b** avec pour sujet **c**.

Exercice 4 - Question 3 (2 points)

```
#!/bin/bash
# script : Q3.sh
for mail in `./Q2.sh $1`
do
    mail -s "$3" $mail <$2
done
```

4. Écrivez un script qui prend 2 arguments :

- le nom **a** d'un fichier au même format que le fichier **elevés.txt**
- le nom **b** d'un fichier texte

et pour chaque ligne du fichier **a** envoie par mail à l'adresse mail de l'étudiant associé un message :

- dont le sujet est la première ligne de **b** (Vous pouvez utiliser la commande **head** pour extraire la première ligne : **head -1 toto** affiche la première ligne du fichier **toto**)
- dont le contenu est celui de **b** sans la première ligne (Vous pouvez utiliser la commande **tail** pour extraire toutes les lignes sauf la première : **tail -n +2 toto** affiche toutes les lignes du fichier **toto** sans la première ligne).

Exercice 4 - Question 4 (2 points)

```
#!/bin/bash
# script : Q4.sh
titre=`head -1 $2`
tail -n +2 $2 > message_tmp
./Q3.sh $1 message_tmp "$titre"
rm message_tmp
```