

# Programmation Système

## TP 1 : Programmation Bash

### Exercice 1

Ecrire un programme shell *repcour* qui affiche le nom de connexion de l'utilisateur et le chemin absolu de son répertoire courant de la manière suivante :

```
$ repcour
mon nom de connexion est : ali
mon repertoire courant est : /home/ali
$
```

### Exercice 2

Ecrire un programme shell *untrois* qui demande à l'utilisateur de saisir une suite de mots constituée d'au moins trois mots et qui affiche sur la même ligne le premier et le troisième mot saisis.

```
Ex : $ untrois
Entrez une suite de mots : un petit coucou de Rodez
un coucou
$
```

### Exercice 3

Ecrire un programme shell *nbmots* qui demande à l'utilisateur de saisir une suite quelconque de mots puis affiche le nombre de mots saisis.

```
$ nbmots
Entrez une suite de mots : un deux trois quatre cinq
5 => 5 mots ont été saisis
$
```

### Exercice 4

Ecrire un programme shell *cp2fois* prenant trois arguments : le premier désigne le nom du fichier dont on veut copier le contenu dans deux fichiers dont les noms sont passés comme deuxième et troisième arguments. Aucun cas d'erreur ne doit être considéré.

### Exercice 5

Ecrire un programme shell *taille* qui prend un nom de fichier en argument et affiche ses droits d'accès et sa taille. On ne considère aucun cas d'erreur.

### Exercice 6

A l'aide de la commande unix *date*, écrire un programme shell *jour* qui affiche le jour courant du mois

```
$ date
dimanche 22 octobre 2006, 18:33:38 (UTC+0200)
$
$ jour
22
$
```

## Exercice 7

Ecrire un programme shell *nbconnect* qui affiche le nombre d'utilisateurs connectés sur la machine locale

## Exercice 8

Ecrire un programme shell *uid* qui affiche l'uid de l'utilisateur. On utilisera la commande unix **id**, la commande interne **set** et la variable prédéfinie **IFS**.

## Exercice 9

En utilisant la commande unix **ls** et le mécanisme de redirection, écrire un programme shell *dansbin* prenant un nom de commande en argument et qui affiche 0 si cette commande est présente dans */bin*, une valeur différente de 0 sinon.

```
$ dansbin ls
0
$ dansbin who
2
$
```

## Exercice 10

On dispose d'un fichier *personnes* dont chaque ligne est constituée du prénom et du genre (*m* pour masculin, *f* pour féminin) d'un individu.

```
Ex : $ cat personnes
Ahmed m
Ali m
Amina f
Khadija f
Najat f
Adil m
$
```

Ecrire un programme shell *tripersonnes* qui crée à partir de ce fichier, un fichier *garçons* contenant uniquement les prénoms des garçons et un fichier *filles* contenant les prénoms des filles.

```
$ tripersonnes
$
$ cat filles
Amina
Khadija
Najat
$
$ cat garçons
Ahmed
Ali
Adil
$
```

## Exercice 11

1. En utilisant les modificateurs de chaînes,
  - a) écrire un programme shell *touslesutil* qui lit le contenu du fichier **/etc/passwd** et affiche le nom des utilisateurs enregistrés.
  - b) modifier ce programme (soit *touslesutilid*) pour qu'il affiche le nom et l'uid de chaque utilisateur.
2. En utilisant les modificateurs de chaînes,
  - a) écrire un programme shell *basenom* ayant un fonctionnement similaire à la commande unix **basename**. Cette commande affiche le dernier élément d'un chemin passé en argument. Il n'est pas nécessaire que ce chemin existe réellement.  
 Ex : \$ **basename** /toto/tata/tutu  
       tutu  
       \$
  - b) si un suffixe est mentionné comme deuxième argument, celui ci est également ôté de l'élément par la commande **basename**.  
 Ex : \$ **basename** /toto/tata/tutu/prog.c .c  
       prog  
       \$

Ecrire un programme *basenom1* qui se comporte de la même manière.

## Exercice 12

Ecrire un programme shell *lsrep* ne prenant aucun argument, qui demande à l'utilisateur de saisir une suite de noms de répertoires et qui affiche leur contenu respectif.

## Exercice 13

1. En utilisant **RANDOM**, écrire un programme *tirage\_flash* qui affiche six entiers différents compris entre 1 et 49.
2. Ecrire un programme *factiter* qui prend un entier  $N$  positif ou nul en argument et affiche sa factorielle  $N!$  Attention aux débordements :  $N$  ne doit pas être trop grand.  
 \$ **factiter** 7  
 5040  
 \$  
 \$ **factiter** 0  
 1  
 \$
3. Ecrire une version récursive *factrecur* du programme précédent.

## Exercice 14

Ecrire un programme shell *tabnoms* qui place dans un tableau les noms de tous les utilisateurs enregistrés dans le système, affiche le nombre total d'utilisateurs enregistrés, puis tire au hasard un nom d'utilisateur.