Université Bordeaux 1 **LST Informatique 2ième Semestre** Lundi 3 mai 2004

Unité d'enseignement INF103 : Utilisation des Systèmes Informatiques
Contrôle de Travaux Pratiques

Documents autorisés. Durée : 2h.

Créez dans votre répertoire d'accueil, un répertoire dont le nom est constitué de vos nom de login :
`mkdir` *login1-login2*.
Placez-vous dans ce répertoire dans lequel vous réaliserez tous les scripts demandés :
`cd` *login1-login2*.
Recopiez le fichier `~idurand/tp-usi/notes.txt` dans votre répertoire :
`cp ~idurand/tp-usi/notes.txt .`
À la fin de la séance, à partir de votre répertoire d'accueil, copiez votre répertoire dans le répertoire `~idurand/tp-usi` :
`cp -r` *login1-login2* `~idurand/tp-usi/`

Exercice 1

Écrire une commande `nbcom` qui prend en paramètre un `nom` de commande et affiche le nombre de commandes de ce nom qui ont été lancées depuis le shell courant.
Exemple :

```
$ ps
  PID TTY          TIME CMD
 1975 pts/1    00:00:01 bash
 2080 pts/1    00:00:28 emacs
 8554 pts/1    00:00:00 emacs
 9499 pts/1    00:00:00 ps
$ nbcom emacs
nombre de emacs lances: 2
```

Exercice 2

Écrire un shell script `renomme` prenant en paramètre une `extension` (`c`, `py`, `sh`, ...) suivie d'une suite de noms de fichiers, éventuellement vide. Ce script renomme chaque fichier `f` en `f.extension` et affiche une trace de la réussite du renommage de chaque fichier.

```
$ ls f[0-9]* g1
ls: g1: No such file or directory
f1  f2
$ renomme txt f[0-9]* g1
f1 renamed
f2 renamed
g1 not renamed
$ ls f[0-9]*
f1.txt  f2.txt
```

On dispose de fichiers contenant pour chaque étudiant, au plus une *description* de la forme
`NO-GROUPE;NOM;PRENOM;NOTE-DS;NOTE-TP`
Par exemple, le fichier `notes.txt` contient

```
01;DUPONT;Charlie;10;15
02;DUPONT;Alfred;04;09
04;MARTIN;Leo;12;15
```

```
02;VAUCEL;Jules;15;18
04;DUPOUY;Linda;14;16
02;VALAIS;Laurence;14;15
```

Exercice 3

Écrire un script `liste-groupe` qui prend en paramètre un `numéro` de groupe et affiche les lignes du fichier correspondant aux étudiants de ce groupe.

Exercice 4

Écrire une fonction `releve-etudiant` qui prend quatre paramètres à savoir le `nom`, le `prénom`, la `note` de DS et la `note` de TP d'un étudiant et qui affiche le relevé de notes de l'étudiant selon le format suggéré par l'exemple suivant :

```
$ releve-etudiant DUPONT Alfred 12 14
DUPONT Alfred
Note de DS: 12
Note de TP: 14
```

Exercice 5

Écrire un script `trie` qui prend en paramètre un fichier de descriptions et affiche les descriptions triées par nom puis par prénom.

Exercice 6

1. Écrire un script `remplace` qui prend en paramètre un fichier de notes et qui affiche sur la sortie standard le contenu du fichier dans lequel le caractère `;` a été remplacé par le caractère `:`.
2. Écrire une deuxième version appelée `remplace2` qui prend en plus en paramètre le caractère à remplacer et le caractère de remplacement. `remplace2 notes.txt ';' ':'` doit donner le même résultat que `remplace notes.txt`.

Exercice 7

Écrire un script `desc2releve` qui prend en paramètre une description `desc` et un répertoire `rep` et qui crée dans le répertoire `rep` un fichier *nom-prenom* contenant le relevé de l'étudiant correspondant à `desc`.
Exemple :

```
$ ls RELEVES/
$ desc2releve '04;MARTIN;Leo;12;15' RELEVES
$ cat RELEVES/MARTIN-Leo
MARTIN Leo
Note de DS: 12
Note de TP: 15
```

Exercice 8

Écrire un script `releves` qui prend en paramètre un nom de répertoire `rep`. Ce script fait appel au script `desc2releve` pour fabriquer dans le répertoire `rep` un relevé pour chaque description qu'il reçoit sur son entrée standard.
Exemple :

```
$ ls RELEVES/
$ releves RELEVES < notes.txt
$ ls RELEVES/
DUPOND-Alfred  DUPOUY-Linda  VALAIS-Laurence
DUPONT-Arthur  MARTIN-Leo    VAUCEL-Jules
```

Ce script fera appel à la commande `read` de `bash` pour lire une à une les lignes sur l'entrée standard.