IUT des Pays de l'Adour
LP RT - ASUR

ASUR4 - TP Système N°5

Programmation Shell

Phrase

- Écrire en shell le programme dont la syntaxe d'appel est la suivante :

```
phrase mot1 ... motN
```

qui affiche VRAI sur la sortie standard si « mot1 » est identique à « motN », et ceci quelque soit le nombre de mots passés en paramètres.
Ex : `phrase toto coucou 1 douze 12 bipbip toto` (affichera VRAI)

Nettoie

- Écrire en shell le programme dont la syntaxe d'appel est la suivante :

```
nettoie suff rep2 rep2 ... repN
```

qui permet d'effacer automatiquement tous les fichiers dont le suffixe est « suff » et qui se trouvent dans un des répertoires passés en paramètres.
Ex : `nettoie txt ~ /tmp ~/tpunix`

Mcp

- Écrire en shell le programme dont la syntaxe d'appel est la suivante :

```
mcp fic rep1 ... repN
```

qui effectue une copie du fichier « fic » sous chacun des N répertoires passés en paramètre (on suppose que les droits d'accès le permettent).
Ex : `mcp ~marcel/toto /tmp ~marcel/reptp ~marcel/reptp2`

Range

- Écrire en shell le programme dont la syntaxe d'appel est la suivante :

```
range depart fic1 fic2
```

qui renomme tous les fichiers qui s'appellent « fic1 » en « fic2 », dans toute l'arborescence issue du répertoire « départ » (pensez à utiliser le « find »).
Ex : `range ~marcel core ajeter`

Cherchedouble

- Écrire en shell le programme dont la syntaxe d'appel est la suivante :

```
cherchedouble user1 ... userN
```

dont le rôle est de chercher dans les répertoires d'accueil des utilisateurs « user1 » à « userN » les fichiers qui portent le même nom que les fichiers du répertoire d'accueil de l'utilisateur qui l'exécute (i.e : vous !)
Ex : `cherchedouble martin durand dupond dubois`

Compte

- Écrire en shell le programme dont la syntaxe d'appel est la suivante :

```
compte depart ress1 ... ressN
```

qui détermine le nombre de ressources existant dans le sous arbre issu du répertoire « depart » et portant les noms passés en paramètres.
Ex : `compte /users toto core tutu`
➔ `il y a 237 toto dans le sous arbre /users`
➔ `il y a 114 core dans le sous arbre /users`
➔ `il y a 12 tutu dans le sous arbre /users`

Compare

Remarque : il est tout a fait possible grâce à la commande exit(N) de faire « remonter » au niveau du shell un compte rendu d'exécution de valeur N (que vous visualiserez alors avec la commande `echo $?` après la fin de votre programme).

- Écrivez donc en shell un programme pouvant être invoqué avec 0 ou 3 paramètres (dans le cas où il est invoqué sans paramètre, il devra les lire sur l'entrée standard).

`compare` ou `compare ch1 ch2 ch3`

Ce programme devra retourner comme compte rendu d'exécution :

0 si les trois chaînes sont identiques,
i (1,2 ou 3) selon que la 1ère, la 2ème ou la 3ème soit différente des deux autres,
4 si les trois chaînes sont différentes
5 si le nombre de paramètres est incorrect

Ex : `compare toto titi tutu`
`echo $?`
➔ 4