

T.D. 7 : Les scripts shell

Exercice 1 La commande `cp` qui permet de copier un fichier vers un autre n'effectue normalement aucun contrôle quant à l'existence éventuelle du fichier destination (si le fichier destination existait, son contenu est perdu).

1. Écrivez un script `ccp` permettant de tester si le fichier destination existe et, en fonction du résultat de ce test, d'autoriser ou d'empêcher la copie.
2. Le script `ccp` suppose que l'opérateur lui a fourni deux paramètres.
 - (a) Que se passe-t-il s'il y a moins de deux paramètres ?
 - (b) Incorporez un contrôle du nombre de paramètres au début du script. (dans le cas où le nombre est insuffisant, un message contenant la manière d'utiliser le programme est affiché).
 - (c) étendez les fonctionnalités du programme de manière à tester toutes les éventualités concernant les deux fichiers transmis en paramètre.

Exercice 2 Écrivez un script qui affiche les fichiers selon une caractéristique passée en paramètre.

- x : affichage des fichiers exécutables
- r : affichage des fichiers lisibles
- x : affichage des fichiers modifiables
- x : affichage des répertoires

Exercice 3 Écrire une séquence shell permettant d'itérer une lecture au clavier jusqu'à l'obtention de la chaîne «oui» ou de la chaîne «non».

Exercice 4 Écrire une commande shell appellable avec un nombre de paramètres compris entre 0 et 3. Cette commande, si elle est appelée avec moins de trois arguments, lira autant de chaînes au clavier que nécessaire. Elle disposera dans tous les cas de trois chaînes. Elle indiquera si les trois chaînes sont identiques, si deux d'entre elles le sont ou si elles sont toutes différentes.

Exercice 5 Écrire une commande shell `dir` qui reçoit comme argument un répertoire et qui affiche tous ses sous-répertoires en format long.

```

$ dir .
drwxr-x--- 2 caron      512 Nov  6 11:37 DEUG
drwxr-x--- 2 caron      512 Nov  6 11:37 FPERM
drwxr-x--- 2 caron      512 Nov  6 11:57 IUP1
drwxr-x--- 2 caron      512 Nov  6 11:38 UNIVP6

```

Exercice 6 Écrire une commande shell `ren` permettant de renommer une liste de fichiers du répertoire de travail. On supposera que les références de fichiers à changer sont données en paramètres de la commande et plus précisément sont tous les paramètres à partir du second. Les nouvelles références seront obtenues à partir du préfixe commun donné en premier argument, le suffixe étant constitué par un entier compris entre 1 et n où n est le nombre de fichiers à renommer. Par exemple l'appel

```
$ ren doc toto truc tata machin
```

renommera respectivement `toto`, `truc`, `tata`, `machin` en `doc1`, `doc2`, `doc3`, et `doc4`. On veillera à ce que la commande ne renomme que les fichiers ordinaires ; on décidera que si la commande n'a qu'un seul paramètre, tout le catalogue de travail est traité. Enfin, on supposera pour simplifier le travail, qu'aucun nouveau nom n'est une référence d'un fichier du catalogue.