

1. RENVERSANT

On se propose d'écrire un script « inverse », qui réécrit sur la sortie standard un fichier en inversant l'ordre de ses lignes. La première ligne devient la dernière ligne, la deuxième ligne devient l'avant-dernière ligne, , la dernière ligne devient la première ligne.

Ce script reçoit en argument le nom du fichier à inverser.

2. COMMANDE WHICH

Quand un exécutable est lancé, le système Linux va chercher en priorité cet exécutable dans le répertoire courant, s'il ne l'a pas trouvé, Linux cherchera dans les répertoires mentionnés dans la variable \$PATH.

```
root@serveur# echo $PATH
```

```
/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin
```

Vous disposez ci-dessous du « man » de la commande which :

NAME

```
which -- locate a program file in the user's path
```

SYNOPSIS

```
which [-as] program ...
```

DESCRIPTION

The which utility takes a list of command names and searches the path for each executable file that would be run had these commands actually been invoked.

The following options are available:

- a List all instances of executables found (instead of just the first one of each).
- s No output, just return 0 if any of the executables are found, or 1 if none are found.

Some shells may provide a builtin which command which is similar or identical to this utility. Consult the builtin(1) manual page.

Écrire un script qui simule le comportement de la commande which.

3. MOT PALINDROME ?

Réaliser un script Shell qui permet de vérifier si un mot est un palindrome. Le mot sera passé en paramètre du script.

Exemple : bob, serres, rotor, radar, laval

4. LES PALINDROMES D'UN FICHIER

Réaliser un script Shell qui permet de trouver tous les palindromes contenus dans un fichier. Le fichier sera passé en paramètre du script.

5. COMMANDE TREE

Écrire une commande « tree » (cette commande existe sous linux) qui affiche l'arborescence d'un répertoire dont le nom est passé en paramètre. Si la commande n'a pas d'argument alors c'est le répertoire courant qui est considéré.

Pour l'affichage, chaque nom de fichier sera précédé de « +-- » et chaque nom de répertoire sera précédé de « \-- ».

Exemple :

```
tree /home/chezmoi
```

```
+-- fichier1
+-- fichier2
\-- rep1
+-- fichier1_rep1
+-- fichier2_rep1
\-- rep1_1
+-- fichier1_rep1_1
+-- fichier2_rep1_1
\-- rep2
+-- fichier1_rep2
+-- fichier2_rep2
```

Conseil : définir une fonction et l'appeler récursivement.

6. DICTIONNAIRE

Sur votre Linux, il existe un fichier words dans le répertoire /usr/share/dict/ contenant tous les mots du dictionnaire, à raison d'un mot par ligne, classés suivant l'ordre alphabétique.

```
./dico.sh 5 i2 u4
```

On cherche un mot de 5 lettres, dont la 2^{ème} lettre est un « i » et la 4^{ème} lettre est un « u »

Ecrire un script « dico », qui permette de retrouver dans le fichier words, des mots qui correspondent aux éléments passés au script.

Par exemple : ./dico.sh 5 i2 u4

On cherche un mot de 5 lettres, dont la 2^{ème} lettre est un « i » et la 4^{ème} lettre est un « u »

Le script renverra par exemple : linux, digue, lieux, minus, sinus, virus

Le script sera toujours appelé avec un paramètre de longueur de mot (dans notre exemple, c'est 5), suivi de plusieurs paramètres indiquant une lettre et sa position (dans l'exemple on a proposé 2 lettres et 2 positions (i2 et u4), mais il peut y en avoir plus).