

TD n°2

Scripts shell

Exercice 1

Ecrire un script shell qui renomme certaines extensions de fichiers dans le répertoire courant :

- fichier.JPG doit devenir fichier.jpeg
- fichier.JPEG doit devenir fichier.jpeg
- fichier.TIF doit devenir fichier.tif

Exercice 2

Ecrire un script shell qui repère les répertoires vides à partir du (des) répertoire(s) passés en paramètres, et demande à l'utilisateur s'il faut supprimer chacun d'eux.

Exercice 3

Une adresse IPv4 est une chaîne de la forme A.B.C.D ou A, B, C et D sont des entiers dans l'intervalle [0,255]. Elle peut également être vue comme un entier 32 bits non signé.

1. Ecrivez une fonction shell `ip2dec` qui, étant donné une adresse IP, donne sa valeur entière en système décimal. Exemples :

```
$ ip2dec 10.0.0.1
167772161
ip2dec 255.255.255.255
4294967295
```

2. Ecrivez la fonction `dec2ip`, réciproque de `ip2dec`.

Rappel : l'expansion arithmétique supporte les opérations '+', '-', '*', '/', et '%' sur les entiers.

Un masque IP est un entier non signé sur 32 bits. Si m est un masque IP, on dit qu'une adresse IP a est masquable par m si $m \times a = 0$.

3. Ecrivez une fonction `bsup` qui, étant donné un masque, donne le nombre total d'adresses IP masquables par ce masque. On conviendra que le masque est codé en hexadécimal sous la forme `0x...` et on pourra utiliser la fonction `printf` (du shell), qui se comporte comme son homonyme en C :

```
$ printf "%d\n" 0xff
255
$ printf "%4.4x\n" 192
00c0
```

4. Ecrivez une fonction `etaler` qui, étant donné un masque et un entier, renvoie la plus petite adresse IP « en masque » et codant cet entier. Exemple :

```
$ etaler 0xf0f0 156    # 156 = 1001|1100, 0xf0f0 =
1111000011110000
2316                  # 2316 = 1001|0000|1100
$
```

5. En utilisant les fonctions précédentes, écrivez une fonction `ipbymask` qui, étant donné un masque, affiche l'ensemble des adresses IP « en masque ».