

### Exercice 1 : Taggons les fichiers

Dans cet exercice vous allez devoir concevoir un système permettant de *tagger* vos fichiers afin de pouvoir les retrouver plus facilement. Il s'agit en fait d'ajouter à volonté une sorte de *méta*-information à vos fichiers. Le principe est simplement de gérer des liens physiques vers vos fichiers dans des répertoires spécifiques (un répertoire par *tag*). Ces répertoires se trouveront sous le répertoire `$HOME/tags`.

**Q 1.** Écrivez un script, nommé `tagger`, qui attend en paramètre une chaîne de caractère (le tag) et un (ou plusieurs) chemin(s) de fichier(s). Ce script doit créer un lien physique vers chacun des fichiers spécifiés dans un répertoire portant le nom du tag dans le répertoire `$HOME/tags`.

**Q 2.** Écrivez un script, nommé `detagger`, qui attend en paramètre une chaîne de caractère (le tag) et un (ou plusieurs) nom(s) de fichier(s). Ce script doit supprimer les liens physiques vers chacun des fichiers qui se trouveraient dans le répertoire portant le nom du tag dans le répertoire `$HOME/tags`.

**Q 3.** Écrivez un script, nommé `nommer`, qui attend en paramètre un numéro d'inode et qui affiche un, **et un seul**, chemin **absolu** vers un des noms du fichier référencé par cette inode, s'il en existe un sous le répertoire `$HOME/tags`.

**Q 4.** Écrivez un script, nommé `lstags`, qui considère que ses arguments constitue une liste de *tags*. Ce script doit afficher un unique chemin vers chacun des fichiers portant un des tags spécifiés.

**Q 5.** Modifiez vos scripts `tagger` et `detagger` de manière à ce que le nombre de *tags* passés ne soit pas limité à un.

### Exercice 2 : Explorons vos fichiers avec le web

Le squelette de programme suivant permet de déléguer le traitement de requête lues sur le port 8080 d'une machine vers un script nommé `lsweb`.

```
1  #!/bin/bash
2
3  PATH=$(cd $(dirname $0) ; pwd):$PATH
4
5  tube=/tmp/tube-$$
6
7  mkfifo $tube
8  while test -r $tube
9  do
10     nc.traditional -q 0 -l -p 8080 < $tube | lsweb > $tube
11 done
12 rm $tube
```

**Q 1.** Écrivez le script `lsweb` de manière à ce qu'il lise, sur son entrée standard, une requête et envoie, sur sa sortie standard, une réponse. La requête et la réponse doivent respecter le protocole HTTP/1.0. Ce script considère que les ressources demandées sont des chemins de fichier (en considérant comme racine votre répertoire principal). La réponse dépend de la nature de la ressource demandée :

- si c'est un répertoire la réponse doit être une page HTML contenant une liste des fichiers de ce répertoire. Cette liste doit contenir des liens HTML vers ses fichiers.
- si ça n'est pas un répertoire ce script la réponse doit être une page HTML mettant en forme les caractéristiques de ce fichier (les informations données par la commande `stat`).

### Exercice 3 : Requêtes citoyennes

Le résultat des votes des lois des 3 dernières législatures de l'assemblée nationale de la république française, sont disponibles en ligne sous la forme de compte-rendu de scrutin web<sup>1</sup>. Ces informations ne sont, *malheureusement*, pas disponibles, *librement*, sous d'autre formes.

Vous allez devoir utiliser des versions textes de ces pages web pour extraire de l'information concernant les différents scrutins. Pour ce TP vous utiliserez les exemples contenus dans les fichiers suivants :

- /home/infoens/beaufils/da2i/0583-dump
- /home/infoens/beaufils/da2i/0695-dump

Étudiez donc le format des ces fichiers, dont le nom contient toujours le numéro de scrutin.

**Q 1.** Écrivez un script, nommé `civilite-depute`, qui prend en paramètre le nom d'un fichier de compte-rendu au format texte puis le prénom et le nom d'un député et qui affiche la civilité à utiliser pour ce député parmi les mots `Monsieur` ou `Madame`.

**Q 2.** Écrivez un script, nommé `vote-depute`, qui prend en paramètre le nom d'un fichier de compte-rendu au format texte suivi du prénom et du nom d'un député. Ce script affiche sur la sortie standard le vote du député pour le scrutin. Le vote affiché sera choisi parmi les mots `OUI` pour le vote *POUR*, `NON` pour le vote *CONTRE*, `ABS` pour une abstention ou `NVP` quand le député n'a pas participé au vote.

---

1. Par exemple pour la législature en cours on trouve ces documents à cette adresse <http://www.assemblee-nationale.fr/14/documents/index-scrutins.asp>