

TP N° 3 : GESTION D'UNE CORBEILLE EN **bash**

Le but de ce TP est de réaliser un gestionnaire de corbeille qui permettra une suppression sécurisée des fichiers de votre compte.

Étape 1 : environnement poubelle

Les futurs scripts de l'utilitaire poubelle devront se trouver dans le répertoire caché `$HOME/.poubelle-util`. Les fichiers supprimés, quant à eux, seront stockés dans un répertoire nommé `$HOME/.poubelle`

1. Écrivez un shell script nommé `$HOME/.poubelle-util/delete` qui aura pour tâche de supprimer les fichiers (mais pas les répertoires pour l'instant) qui lui sont passés en paramètres et de les placer dans le répertoire `$HOME/.poubelle`. Le répertoire `$HOME/.poubelle` sera créé automatiquement par le script s'il n'existe pas.
2. Faites en sorte que `delete` puisse être exécuté à partir de n'importe quel endroit du système de fichiers.
3. Afin de pouvoir restaurer les fichiers effacés, il est nécessaire qu'à chaque suppression, on maintienne un fichier d'index nommé `INDEX` (que l'on placera dans `$HOME/.poubelle-util`) qui associera à chaque fichier son répertoire d'origine. Par exemple si l'on supprime les fichiers `fic.txt` et `/opt/fic2.txt`, le fichier `INDEX` contiendra les lignes :

```
fic.txt:/home/user1
```

```
fic2.txt:/opt
```

Modifiez le script `delete` en ce sens. **Aide** : vous pourrez utiliser la commande `readlink -f` pour récupérer le chemin absolu d'un fichier.

Si jamais on supprime un fichier dont le nom existe déjà, le script doit demander si l'on souhaite remplacer dans la poubelle le fichier préalablement supprimé ou pas.

4. Écrire un script `ls-poubelle` qui affiche l'ensemble des noms des fichiers de la poubelle avec leur répertoire d'origine. L'affiche sera de la forme suivante :

```
La poubelle contient 2 fichiers :
```

```
fic.txt de répertoire d'origine /home/user1
```

```
fic2.txt de répertoire d'origine :/opt
```

Si la poubelle est vide, on affichera le message : « la poubelle est vide ».

5. Écrire un script `vide-poubelle` qui vide la poubelle.
6. Écrire un script `restaure` qui à partir d'un nom de fichier restaure le fichier correspondant dans son répertoire d'origine. Si le fichier à restaurer existe déjà on affiche un message demandant si l'on souhaite écraser le fichier existant.

Étape 2 : compression des fichiers supprimés

Les fichiers qui seront supprimés vont maintenant être stockés sous une forme compressée dans la poubelle à l'aide de l'utilitaire `gzip`.

Modifiez les scripts `delete`, `ls-poubelle`, `vide-poubelle` et `restaure` en ce sens.

Étape 3 : Suppression de répertoires

On veut pouvoir supprimer des répertoires et les placer archivés et compressés dans la poubelle. Vous utiliserez pour ça les utilitaires `tar` et `gzip`.

Modifiez les scripts `delete`, `ls-poubelle`, `vide-poubelle` et `restaure` en ce sens ; `ls-poubelle` devra indiquer parmi les éléments affichés lesquels sont des fichiers et lesquels sont des répertoires.

Étape 4 : Contrôle de cohérence de la poubelle

On souhaite pouvoir vérifier que le contenu de la poubelle est cohérent avec sa description faite dans le fichier `INDEX`. Écrire un script `poubelle-ok` qui détermine si des éléments de `INDEX` ne sont pas dans la poubelle et si des éléments de la poubelle ne sont pas dans le fichier `INDEX`.