INF203 - Travaux pratiques, séance 4
Scripts shells : redirections, variables, conditions

1 Une commande utile pour la suite : `wc`

La commande `wc` permet de compter les caractères, les mots, les lignes d'un fichier ASCII. Lisez son manuel et repérez l'option à utiliser pour compter les lignes.
[a] Quelle est cette option ? D'après le manuel, quels caractères la commande `wc` compte-elle pour compter les lignes du texte ? ■
Souvenez-vous de la manière de désigner tous les fichiers du répertoire courant se terminant par `.txt`. Utilisez `wc` (avec la bonne option) sur tous les fichiers se terminant par `.txt` de votre répertoire `[TP4]`.
[b] Quelle commande avez-vous saisi ? Combien y-a-t-il, au total, de lignes de texte dans votre répertoire ? ■

2 Gestion de corbeille

Le but de cet exercice est de gérer une corbeille permettant la suppression de fichiers avec restauration possible de la dernière version. La corbeille sera un sous-répertoire `Corbeille` du répertoire principal de l'utilisateur. Elle est gérée à l'aide de 2 scripts shell appelés `supprime.sh` et `restaure.sh` :

- `supprime.sh` prend un nom de fichier en argument, et le déplace dans la corbeille ;
- `restaure.sh` prend un nom de fichier (sans chemin) en argument et récupère la version la plus récente de ce fichier présente dans la corbeille.

Pour que cette restauration soit possible, les noms des fichiers de la corbeille sont suffixés, lors de la suppression, par un numéro représentant le nombre de fichiers du même nom présents dans la corbeille. Le plus récent est donc celui de numéro le plus élevé. On utilisera un script intermédiaire `nombre_dans_corbeille.sh` qui prend un nom de fichier `fich` (sans chemin) et qui affiche le nombre de fichiers de la forme `fich@`<numero> présents dans la corbeille.

IMPORTANT. On supposera que le nom d'un fichier dans la corbeille (sans son numéro) n'est jamais le préfixe du nom d'un autre fichier dans la corbeille. Par exemple, il n'y aura jamais dans la corbeille à la fois `ploum` et `ploumplam`. Vous n'avez pas à le vérifier, et on ne demande pas que vos scripts respectent cette spécification.

Exemple :

commandes successives	contenu corbeille résultant	commentaires
	vide	
`./supprime.sh ploum`	`ploum@1`	ploum est supprimé du répertoire courant
`./supprime.sh ploum`	`ploum@1 ploum@2`	La version la plus récente de ploum est ploum@2 L'ancienne version est conservée.
`./nombre_dans_corbeille.sh ploum`	`ploum@1 ploum@2`	affiche 2
`./restaure.sh ploum`	`ploum@1`	Restauration de ploum (version 2) dans le répertoire courant
`./supprime.sh ../TP25/essai.c`	`ploum@1 essai.c@1`	Nom de fichier sans chemin dans la corbeille

2.1 Écriture de `nombre_dans_corbeille.sh`

À l'aide d'une ou plusieurs redirections bien choisies, écrivez le script `nombre_dans_corbeille.sh`. Vous vérifierez qu'il comporte exactement un argument. En revanche, vous supposerez que ce script est appelé avec un argument qui est un nom de fichier sans chemin. *Rappel :*

1. la commande `wc -w` sans argument affiche à l'écran le nombre de mots de l'entrée standard.
2. Si le fichier `ploum` n'existe pas, la commande `ls ploum` affiche un message d'erreur. Pour ne pas afficher ce message d'erreur, il suffit de taper, à la place, `ls ploum 2> /dev/null` .

2.2 Écriture de `supprime.sh`

Écrivez le script `supprime.sh`, afin qu'il prenne un nom de fichier en argument et :

— qu'il crée le répertoire `Corbeille` dans le répertoire principal de l'utilisateur, sans message d'erreur dans le cas où celui-ci existe déjà ;

— qu'il déplace le fichier donné en argument dans la corbeille ;

— en retirant son chemin si nécessaire

— et en suffixant le nom avec le numéro approprié, qu'il est possible de déterminer en faisant appel à `nombre_dans_corbeille.sh`.

En outre, vous vérifiez au début de ce script, qu'il y a exactement un argument et que cet argument est un fichier.

2.3 Écriture de `restaure.sh`

Écrivez le script `restaure.sh` qui prend un nom de fichier sans chemin en argument, et déplace la version la plus récente de ce fichier (si elle existe) depuis la corbeille vers le répertoire courant. En cas de problème, affichez un message d'erreur. En outre, vous vérifiez au début de ce script, qu'il y a exactement un argument et que cet argument est bien « sans chemin » (à l'aide de la commande `basename`). *Attention :* il ne faut pas vérifier que c'est un fichier (a priori, il n'existe pas).

2.4 `restaure.sh` avec option

Modifiez le script `restaure.sh` pour qu'il prenne un ou deux arguments (dans le cas contraire, il faudra un message d'erreur). Dans le premier cas, le comportement du script sera identique à la version précédente. Dans le second cas, le second argument devra être un répertoire (à vérifier). Ainsi, la version la plus récente du fichier, toujours sans chemin, en premier argument (si elle existe) devra être déplacée de la corbeille vers ce répertoire.

2.5 `supprime.sh` avec plusieurs arguments

Modifier le script `supprime.sh` pour qu'il puisse prendre plusieurs fichiers en arguments, et dans ce cas les place dans la poubelle suivant la méthode précédente.

[c] Joignez le texte des trois scripts `nombre_dans_corbeille.sh`, `supprime.sh` et `restaure.sh` à votre compte-rendu. ■

Exercice complémentaire :

3 Une autre gestion de corbeille

Cette nouvelle corbeille sera stockée dans le sous-répertoire `Corbeille2` de votre répertoire principal. L'idée est de restaurer un fichier à l'endroit exact où il a été supprimé. Ainsi, on gardera trace du chemin absolu où le fichier été initialement stocké dans un fichier caché `.index` de la corbeille. Chaque fichier placé dans la corbeille sera maintenant suffixé `@<numero>` où `<numero>` est un numéro à usage unique. Le dernier numéro utilisé sera stocké dans le fichier caché `.num` de la corbeille. L'utilisation de la corbeille se fera au

moyen de trois scripts : `supprime_v2.sh`, `restaure_v2.sh` et `liste.sh`. Nous aurons besoin également d'un script intermédiaire appelé `absolu.sh`.

`absolu.sh :` Ce script prend en paramètre un fichier (désigné par un chemin absolu ou relatif) et affiche le chemin absolu de ce fichier. Par exemple, si dans le répertoire `TP4`, on exécute `./absolu.sh ../TP1/Candide_chapitre1.txt`, alors le script affiche

`/home/d/devismes/INF203/TP1/Candide_chapitre1.txt`

Vous vérifierez qu'exactement un argument est en paramètre de ce script et qu'il est bien un fichier. Le code de retour du script sera 0 uniquement en cas de succès.

Indications : utilisez les commandes `echo`, `dirname`, `basename`, `cd`, `pwd` et `exit`.

`liste.sh :` Ce script affiche le contenu du fichier `.index`, si ce dernier existe dans le sous-répertoire `Corbeille2` de votre répertoire principal.

`supprime_v2.sh :` Ce script réalise les étapes suivantes

- Il vérifie qu'il y a exactement un argument et qu'il s'agit d'un fichier.
- Il vérifie l'existence du sous-répertoire `Corbeille2` de votre répertoire principal. S'il n'existe pas, il le crée avec, à l'intérieur, un fichier vide `.index` et un fichier `.num` contenant uniquement le nombre 0.
- Il déplace le fichier `fich` en argument dans la corbeille en le renommant `fich_c@<numero>`, où `fich_c` correspond à `fich` sans son chemin et `<numero>` est égal à 1 plus le numéro contenu dans `.num`.
- Il met à jour le fichier `.num` en incrémentant son ancienne valeur.
- Enfin, il ajoute une ligne au fichier `.index` de la forme `fich_c@num:absolu`, où `absolu` est le chemin absolu vers `fich`.

Supposons que le premier appel à `supprime_v2.sh` soit `./supprime_v2.sh ../TP1/Candide_chapitre1.txt`. Après l'appel, la corbeille contient `Candide_chapitre1.txt@1`, le fichier `.num` contient 1 et le fichier `.index` contient la ligne `Candide_chapitre1.txt@1:/home/d/devismes/INF203/TP1/Candide_chapitre1.txt`.

Indications : utilisez `absolu.sh` et les commandes `mkdir`, `touch`, `echo`, `basename`, `cat`, `expr` et `mv`.

`restaure_v2.sh :` Ce script prend en paramètre un numéro `<numero>` (on testera uniquement l'existence d'exactement un paramètre) et restaure le fichier avec le suffixe `@<numero>` (s'il existe) à sa position d'origine. Par exemple, suite à la création de `Candide_chapitre1.txt@1` dans la poubelle, un appel à `restaure_v2.sh 1` déplacera et renommera `Candide_chapitre1.txt@1` en

`/home/d/devismes/INF203/TP1/Candide_chapitre1.txt`

Enfin, la ligne le concernant (c'est-à-dire, la ligne contenant `@1:`) dans `.index` sera supprimée.

Indications : utilisez les commandes `echo`, `grep` et `mv`.

[d] Joignez le texte des quatre scripts `supprime_v2.sh`, `restaure_v2.sh`, `liste.sh` et `absolu.sh` à votre compte-rendu. ■