



mathieu.barcikowski@sfr.fr

thibault.caron@univ-savoie.fr

jean-marc.duchamp@univ-savoie.fr

Sujet TP n°2

Objectifs : gestion d'un répertoire « poubelle »

1. Présentation

La commande Linux « rm » a pour effet de supprimer définitivement les fichiers avec le cas échéant la possibilité de demander une confirmation (option -i). Toujours est il que les fichiers supprimés ne peuvent pas être récupérés.

Dans le cadre de ce TP on se propose de définir un script réalisant la fonction « del » qui permet de transférer les fichiers à effacer dans un répertoire poubelle « .Trash » accessible depuis le répertoire « home » de chaque utilisateur.

Un second script « GestPoub » avec un menu interactif permettra de visualiser, restaurer ou supprimer définitivement les fichiers dans ce répertoire.

Une troisième partie assurera le transfert automatique de fichiers spécifiques tels que les fichiers temporaires (*.tmp) dans le répertoire poubelle à intervalle de temps réguliers.

Les scripts pourront contenir des structures de choix conditionnels (if else fi, case esac) et des boucles (while, until). L'utilisation de telles structures nécessitera la recherche des informations nécessaires à leur mise en œuvre.

2. Création d'un répertoire « poubelle » sous Unix

2.1. Script de commande « del »

La commande « del » déplace les fichiers spécifiés dans un répertoire « .Trash » accessible dans le « home directory » de chaque utilisateur.

Le script doit vérifier que :

- ce répertoire existe ou le créer le cas échéant
- la commande « del » contient au moins un argument (noms des fichiers à supprimer)
- réalisation le déplacement du ou des fichiers spécifiés

Le déplacement peut se faire si

- le ou les fichiers spécifiés existent
- la poubelle ne contient pas de fichiers de même nom

Proposer un contenu pour le script « del »

2.2. Sauvegarde du chemin d'origine

Lors du transfert des fichiers dans le répertoire « .Trash » ou suppression provisoire, il est nécessaire de sauvegarder l'emplacement d'origine des fichiers supprimés.

Pour cela on propose de créer un fichier caché qui contiendra le répertoire d'origine de chaque fichier supprimé. Par exemple la suppression de « Monfichier.txt » dans le répertoire \$HomeDirectory/MesDocuments/Textes doit générer la création d'un fichier texte « .Monfichier.rep » dans le répertoire « .Trash » qui contiendra la ligne « \$HomeDirectory/MesDocuments/Textes ».

Modifier votre script pour mettre en place la sauvegarde du chemin d'origine.

3. Manipulation des fichiers dans le répertoire « poubelle »

3.1. Menu interactif

Le script de commande « GestPoub » doit permettre de manipuler le contenu du répertoire .Trash à l'aide d'un menu interactif, tel que celui décrit ci-dessous

Gestion de la poubelle

- 1) Afficher contenu de la poubelle
- 2) Restaurer d'un ou plusieurs fichiers
- 3) Supprimer d'un ou plusieurs fichiers
- 4) Vide la poubelle
- 5) Fin gestion poubelle

« Afficher le contenu » présente à l'écran l'ensemble des fichiers supprimés (à l'exclusion des fichiers de suivi du répertoire d'origine « *.rep »).

« Restaurer un ou plusieurs fichiers » replace les fichiers dans leur répertoire d'origine. Il est nécessaire de prévoir la saisie d'un identifiant du ou des fichiers à restaurer. Par exemple la saisie « *.C » restaurera tous les fichiers C dans leur répertoire.

La suppression définitive d'un ou plusieurs fichiers implique également la saisie d'un identifiant. Par exemple la commande « r*.exe » supprimera tous les programmes débutants par « r ».

Elaborer un script réalisant ce menu interactif et les fonctions associées à ce menu

3.2. Option de commande

La syntaxe générale d'une commande UNIX est la suivante
« Commande [-option] arguments »

La commande UNIX « getopts » permet d'associer aux scripts une gestion analogue à celle des commandes UNIX avec des options et des arguments.

L'exécution du script « GestPoub » sans option lancera le menu interactif précédent. Les options -A, -R, -S, et -V correspondront respectivement aux quatre choix du menu précédent.

Par exemple la commande « ./GestPoub -R MonFichier.txt » restaurera le fichier spécifié dans son répertoire d'origine.

Modifier le script « GestPoub » pour autoriser la manipulation des options -A, -R, -S, et -V et de leurs arguments.

4. Tâches automatisées

La commande **crontab** permet de visualiser (option -l) ou d'éditer (option -e) une table de tâches planifiées. Ces tâches seront effectuées à heures fixes.

Pour éditer la table des tâches planifiées du demon « cron », il suffit d'exécuter la commande « crontab -e ». Cette commande a pour effet de lancer l'éditeur vi présentant la table actuelle (donc aucune s'il s'agit du premier lancement de crontab).

Relever le PID du processus cron ainsi que le PID du processus père.

Modifier le menu pour interactif de GestPoub pour permettre la programmation de la vidange de la poubelle périodiquement (par exemple toutes les semaines) à l'aide des tâches planifiées « cron »

5. Compression

Pour gagner de la place on se propose de compresser automatiquement tous les fichiers temporairement supprimés en fin de journée.

Ajouter cette fonction à la liste des tâches automatisées.

Lors de la restauration il est nécessaire de décompresser le ou les fichiers restaurés.

