

TP Programmation Shell

Remarques: Lorsque vous utilisez des commandes ``dangereuses" telles que rm, mv, cp ou des redirections hasardeuses :

*Précédez toutes ces commandes par echo, ainsi votre script affichera ce qu'il va faire avant l'exécution.

* Ou encore, utilisez les versions interactives (cp -i, mv -i et rm -i), le script vous demandera une confirmation avant de faire les manipulations.

*Ou testez votre script dans /tmp après avoir créé quelques fichiers d'essai.

Exercice1

En utilisant la commande find, créez une commande **script-backup** permettant de copier tous les fichiers de votre *home directory* dont le nom se termine par un ~ (fichiers backup) dans un répertoire /tmp/BACKUP.

Le script permet aussi de créer le répertoire /tmp/BACKUP s'il n'existe pas.

Exercice2

L'objectif de cet exercice est de créer une commande **script-trash** qui déplacera les fichiers dans un répertoire ~/ma_corbeille au lieu de les effacer.

1. Créez un script « script-trash » qui permet de créer un répertoire ~/ma_corbeille si ce dernier n'existe pas.

2. Testez sur les options passées avec le script-trash :

*L'option -c (script-trash -c) affiche la taille du contenu du répertoire ma_corbeille.

On utilisera la commande du -sk.

*L'option -e permettant de vider le contenu de ma_corbeille.

*L'option -h affiche une aide pareil dans le cas ou on lance le script sans argument:

On utilisera :

cat<<EOF

usage : script-trash [-e | -h | -c] [file1 file2 ... file3]

EOF

3. Le script-trash permet aussi de faire une suite de commandes nécessaire pour déplacer tous les fichiers passés en argument sur la ligne de commande dans le répertoire ma_corbeille.

On utilisera une boucle for et \$*. On tapera par exemple: script-trash file1 file2 pour déplacer file1 et file2 dans ma_corbeille.

Exercice 3

L'objectif de cet exercice est de créer une commande **script-kill** qui demande un nom du programme à tuer à l'utilisateur par la suite exécute le kill.

Le script n'exécute le kill si seulement si l'utilisateur fait une confirmation. Pour cela le script affiche un message « Voulez vous supprimer le processus ? ».

On utilisera les commandes tr -s ' ', cut, kill, read, ps -fA, grep, grep -v et echo pour la recherche du programme.

1- Lire les pages de man des commandes utilisées.

2- La commande affichera:

Donnez le nom du processus que vous voulez supprimer?

Vous rentrerez un nom de processus, par exemple firefox et tous les processus firefox seront supprimés.

3- Quels sont les grep qu'il faut effectuer sur la sortie de ps -fA pour ne conserver que le processus que l'on souhaite tuer.

Utilisez la variable \$USER.

Stockez la liste de PID des processus à supprimer dans la variable PID.

Le script affiche le message : « Le PID du processus à tuer est:xxx »

1. Écrire la commande. N'oubliez pas de vérifier que la variable \$PID n'est pas vide et aussi d'afficher un message de confirmation avant d'effectuer le kill.