

EXERCICE 1: Premier script

Écrire un script contenant la séquence de commandes suivante:

- créer un répertoire temporaire C_FILE,
- créer dans ce répertoire le fichier c_liste contenant la liste des noms des fichiers du répertoire de travail dont le nom se termine par l'extension .c ou .o,
- recopier le fichier correspondant au dernier nom de cette liste dans C_FILE
- lister le contenu du fichier c_liste

Rendre exécutable le fichier contenant ce script. Vérifier qu'il fonctionne bien, en contrôlant l'existence du répertoire C_FILE et d'un fichier dans ce répertoire dont le nom porte l'extension .c ou .o.

EXERCICE 2: Commande laquelle

Écrire une commande *laquelle* à un argument qui, en explorant le path, retrouve l'emplacement de cet l'exécutable donné en argument dans l'arborescence.

Exemple d'exécution: *laquelle* grep doit vous retourner « /bin/grep » et *laquelle* « gcc » vous retourne /usr/bin/gcc. Cette commande est la commande *which* simplifiée.

EXERCICE 3: Commande suffixer

1) Écrire une commande *suffixer* avec deux arguments, le premier est une chaîne de caractères, considérée comme un suffixe, le deuxième est un nom de fichier ou de répertoire. Cette commande ajoute le suffixe à la fin du nom du fichier ou du répertoire.

Pour simplifier l'écriture du script, on suppose que le nom du répertoire est donné sans le / terminal.

L'exécution de la commande : *suffixer* chaine nom

... a pour effet de renommer le fichier ou le répertoire <nom> en <nom>.<chaîne>

Par exemple : > *suffixer* old prg.c ... renomme le fichier prg.c en prg.c.old

Exemples d'utilisation :

> *suffixer*

> usage : *suffixer* suffixe nom_de_fichier_ou_de_repertoire

> *suffixer* old toto.c

> *suffixer*: toto.c inexistant

Conseil : Lors de la mise au point d'un script-shell, quel qu'il soit, il est intéressant de faire afficher les actions effectuées au fur et à mesure de leur exécution. Cet affichage est possible en mettant l'instruction « set -x » au début du script.

2) Modifier cette commande de façon à ce qu'elle s'applique à une liste de noms de fichiers : le suffixe (en premier argument) est ajouté à chacun des fichiers (suite des arguments).

Par exemple : la commande suivante permet d'ajouter le suffixe `old` aux trois fichiers droits, toto et ville1.txt, s'ils existent :

> *suffixer* old droits toto ville1.txt

> *suffixer*: toto inexistant

La commande *suffixer* poursuit son exécution même si l'un des fichiers de la liste n'existe pas.

EXERCICE 4 : Commande *chsuff*

1) Écrire une commande *chsuff* à deux arguments, le premier est une chaîne de caractères représentant un suffixe ou le symbole « - » , le deuxième est un nom de fichier ou de répertoire.

L'exécution de la commande : *chsuff* chaîne nom

... a pour effet : de remplacer le suffixe de <nom> par <chaîne>, si <nom> comporte un suffixe, ou de renommer l'objet par <nom>.<chaîne> sinon.

L'exécution de la commande : *chsuff* - nom

... a pour effet de supprimer le suffixe de <nom>, s'il existe.

Exemples d'utilisation :

> *chsuff* old prog.c : renomme prog.c en prog.old (changement de suffixe)

> *chsuff* xx prog : renomme prog en prog.xx (ajout d'un suffixe)}

> *chsuff* - prog.old : renomme prog.old en prog (suppression du suffixe)

2) Reprendre la commande de façon à ce qu'elle s'applique à une liste de noms de fichiers ou de répertoires.

EXERCICE 5 : commande *supprimeDouble*

Écrivez un script *supprimeDouble* qui prend un répertoire en argument et y supprime tous les fichiers doublons. Pour cela, utilisez la commande *cmp*, et commencez par tester le script sans utiliser *rm* mais simplement en affichant les noms des fichiers doublons trouvés.