

TP3 Shell, 1A apprentissage 2015/2016

EXERCICE 1: commandes diverses

Écrire une commande *audit* qui regarde dans votre home les fichiers exécutables par tous les utilisateurs et les affiche (utilisez *find* et *grep*). Indication : utilisez le paramètre -type de *find* pour ne prendre que les fichiers et pas les répertoires.

Écrire une commande *vire-ligne-vide* permettant de lister le contenu d'un fichier passé en argument de la commande, sans les lignes vides et en les numérotant. Indication : voir le manuel de *grep* et *cat*.

Écrire une commande *grep-rec* qui recherche dans l'arborescence située sous le premier argument, toutes les lignes qui vérifient l'expression située en second argument, dans le fichier passé en troisième argument.

Par exemple : `grep-rec . "^#" "*.c"`

Indication : voir le manuel de *xargs*.

EXERCICE 2: affiche-session

Écrire un script *affiche-session* qui affiche la liste de tous les fichiers modifiés dans la journée courante. Écrivez une version avec *sed*. Indication : commencer en mettant la date en dur dans votre script, puis voir le manuel de *date*.

EXERCICE 3: affiche

- 1) Écrire une commande *affiche* qui liste les noms des fichiers du répertoire courant tels que :
 - ils possèdent le suffixe `.c`
 - ils contiennent la ligne : `#include <stdio.h>`
- 2) Améliorer la commande afin de vérifier que la ligne `#include <stdio.h>` n'est présente qu'une seule fois et exclusivement sur la première ligne du fichier.

EXERCICE 4 : replace

Écrire une commande *replace* qui est un *sed* simplifié, à utiliser dans les cas simples. La syntaxe de cette commande est la suivante : `replace chaine1 chaine2 fich`.

Elle a pour effet de remplacer toute occurrence de `chaine1` par `chaine2` dans le fichier `fich`. Faites en sorte que cette commande fonctionne aussi dans le cas d'une liste de fichiers (par exemple: `replace chaine1 chaine2 '*.txt'`).

EXERCICE 5 : Petit utilitaire de pense-bête.

L'objectif est d'écrire un utilitaire permettant de stocker et retrouver aisément des bribes d'informations dans un fichier. Par défaut, il stocke les informations passées en argument dans le fichier `~/pensebete`. Si aucun argument n'est passé, il lit les informations au clavier et les stocke dans le fichier avec la commande `cat - >> ~/pensebete`.

Si le premier argument est `-c`, le script permet de rechercher des informations dans le fichier. L'argument suivant est alors le motif à chercher avec `grep`.