TP noté 2023 : Système d'exploitation

Durée : 1h30

Documents autorisés : Le poly système uniquement

Enseignante : Fetia Bannour

Exercice 1

(a) Donnez une commande qui permet de calculer le nombre total de lignes que font les fichiers .c ou .h dans une arborescence /tmp donnée.

(b) Indiquez ce que fait la commande suivante:

```
sh> grep $USER /etc/passwd |sed -e "s/[^:]*:[^:]*:\([^:]*\):\([^:]*\).*/\2\t \1/"
```

(c) Indiquez ce que fait la commande suivante:

```
sh> find . -type f -size +20M -mtime +25 -exec mv \{\} ~/archive/ \;
```

Exercice 2

Considérons le script shell fonctions.sh ci-dessous.

```
fonction1()
{
    f=$1
    cat $1 | tr -cs '[:alpha:]' '\n' | tr 'A-Z' 'a-z' | sort | uniq -
    c | sort -nr
}
fonction2()
{
    f=$1
    sed -n '3~5p' $f
}
```

fonctions.sh

- (a) Indiquez ce que fait la builtin commande fonction1.
- (b) Indiquez ce que fait la builtin commande fonction2.

En utilisant ce script, écrivez un script shell script.sh qui se lance avec 4 arguments: sh script.sh file1 file2 output1 output2

Le script suivra les usages standards d'Unix. Son comportement est le suivant:

- (c) Il vérifie le nombre d'arguments: S'il n'a pas 4 arguments, il affiche un message d'erreur standard (stderr) et se termine avec un statut d'erreur.
- (d) Pour chacun des arguments file_i, il vérifie si le fichier file_i est accessible en lecture. Si l'un de ces fichiers n'est pas accessible en lecture, il affiche un message d'erreur standard (stderr) et se termine avec un statut d'erreur.
- (e) Pour chacun des arguments output_i, il vérifie si le fichier output_i est accessible en écriture. Si l'un de ces fichiers n'est pas accessible en écriture, il affiche un message d'erreur standard (stderr) et se termine avec un statut d'erreur.
- (f) Il écrit dans le fichier output1 les éléments suivants un par ligne :
 - Le mot le plus fréquent dans tous les fichiers $file_i$.
 - Le nombre d'occurrence maximale de ce mot.
 - Le mot le moins fréquent dans tous les fichiers file_i.
 - Le nombre d'occurrence maximale de ce mot.
- (g) Il lit un entier N sur le flux standard d'entrée (stdin), et vérifie que 0<N<nombre de mots distincts dans le fichier file₁, sinon il affiche un message d'erreur et échoue.
- (h) Il écrit dans la suite du fichier output1 et sur une même ligne.
 - les № mots les plus fréquents dans le fichier file₁, séparés par des espaces.
- (i) Il lit un entier X et un entier Y sur le flux standard d'entrée (stdin), et vérifie que 0 < X < Y, sinon il affiche un message d'erreur et échoue.
- (j) Il écrit dans le fichier output2:
 - le contenu des lignes de file₂ en remplaçant (sans tenir compte de la casse) toutes
 les occurrences de l'URL http://www.ecole.fr/ par https://www.ensiie.fr/,
 et TD par TP, et ce en utilisant une seule commande shell.
 - le contenu des lignes commençant par le mot ${\tt Unix}$ (sans tenir compte de la casse), allant de la ligne numéro ${\tt X}$ à la ligne numéro ${\tt Y}$.
 - les lignes non vides de file₂, excluant les lignes allant de la ligne numéro X à la dernière ligne de file₂.
 - chaque 5ème ligne de file₂ commençant par la 3ème ligne en remplaçant la première occurrence de shell par bash dans ces lignes.
- (k) Il écrit sur la sortie standard (stdout) le contenu des fichiers output1 et output2.
- (l) Il crée un archive [votre-nom].tar dans le répertoire actuel contenant les fichiers output1 et output2.

Donnez le code shell de toutes ces parties du script script.sh en se servant des deux fonctions du script fonctions.sh, et en créant d'autres au besoin.