

TP N°9 : SED

L'éditeur de flot **sed** lit les lignes du fichier une à une, leur applique des commandes, et envoie le résultat sur la sortie standard.

Quelques commandes:

d	détruit la ligne ; lit la suivante
p	imprime la ligne
s/anc/nouv/f	substitue anc par nouv dans f si f = g remplace toutes les occurrences ; f = p, imprime ; f = w <i>fic</i> écrit dans fic
y/str1/str2/	remplace chaque caractère de <i>str1</i> par le correspondant de <i>str2</i> (pas d'intervalles)
=	imprime le numéro de la ligne courante
!cmd	exécute la commande <i>cmd</i> si la ligne n'est pas sélectionnée
w fichier	écrit la ligne dans le fichier fichier
r fichier	lit fichier, et copie son contenu en sortie
a\ texte	ajoute des lignes à la sortie jusqu'à la première ne se terminant pas par \
c\ texte	remplace des lignes comme pour a
i\ texte	insère le texte suivant avant la prochaine sortie

Options

L'option **-n** permet de ne pas afficher la source lors des modifications

L'option **-e** instruction permet de spécifier plusieurs instructions

L'option **-i** modifie le fichier paramètre plutôt que d'envoyer vers la sortie standard

Script: Les commandes peuvent être regroupées dans un fichier :

```
sed -f fichier.sed fichier.txt
```

Filtre: **sed** peut être utilisé pour filtrer la sortie d'une commande:

```
ls -F .. | sed "s/\\/((répertoire)/g"
```

COMMANDES SED

Créer le fichier **argent.txt** contenant le texte suivant :

J'aime mon argent. J'aime mes EUROS
Je répète que j'ai besoin de mon argent pas de mon argenterie...
j'utilise mon argent et la monnaie
La banque veut mon argent, tout mon fric !

Donner les commandes sed qui affichent le contenu du fichier **argent.txt** en réalisant l'opération spécifiée :

- Remplacer mon par ton partout
- Afficher uniquement la première ligne *proposer 2 solutions différentes*
- Ne pas afficher les lignes 2 et 3 *proposer deux solutions, l'une utilisant la négation*
- Afficher les numéros des lignes commençant par J ou j (*uniquement les N°*)
- Ajouter un point à la fin de chaque ligne qui se termine par une lettre
- Remplace J' et Je par Il en début de ligne
- Remplacer toutes les voyelles minuscules par une étoile * *proposer deux solutions*
- Débuter toutes les lignes ne commençant pas par un J, par un T
- Copier les lignes commençant par J dans le fichier personnel.txt
- Encadrer tous les occurrences du mot argent par des chevrons : <<argent>>

Créer le fichier **os.txt** contenant le texte suivant :

j'aime linux
je rêve de linux
linux, linux, toujours linux
je déteste windows

Le texte résultant du traitement sed devra être le suivant :

J'aime Windows.
Je rêve de Windows.
Windows, Windows, toujours Windows.
Je déteste Linux...

(notez que Linux est écrit avec une majuscule dans le résultat)

Donner la succession de commandes sed transformant le fichier **os.txt**

FILTRES SED

Filtrer la sortie de la commande `who` pour ne conserver que les noms d'utilisateurs et l'heure de connexion, p.ex.: `util1 14.30`

Filtrer la sortie de la commande `date` de telle sorte que lundi 7 novembre 2018, 15:57:00 (UTC+0200) s'affiche comme suit :

```
lundi-7-novembre  
15:57:00
```

SCRIPTS SED

Donner le script ***instructions.sed*** qui supprime la première ligne et les commentaires d'un script.

Écrire le **script *nettoyer.sed*** qui pour le fichier passé en paramètre :

- retire la ponctuation (,;.?)
- retire les lignes vierges (ou ne contenant que des espaces)
- retire les accents sur les e (é->e,è->e,e,ê->e)
- remplace par une * le premier chiffre de tous les nombres

Exécuter le script ***nettoyer.sed*** :

- sur `os.txt`
- en filtrant la page man de la commande `mkdir`
- sur tous les fichiers portant l'extension `.txt` de l'arborescence de l'utilisateur `user`

SCRIPT BASH

Ecrire un script ***commenter.sh*** qui commente un bloc de lignes d'un script passé en paramètre, en insérant des `#` en début de ligne.

```
./commenter.sh 10 15 script.sh # commente les lignes 10 à 15 du script script.sh
```

QUESTIONS

1. Ecrire un script qui renomme les fichiers en remplaçant les espaces par des _
2. Donner la ligne de commande qui affiche le fichier, en encadrant la première occurrence de nombre de chaque ligne par des étoiles
3. Retirer la ponctuation d'un texte
 - a) avec une expression régulière simple
 - b) avec une expression régulière étendue
4. Créer un fichier *liste* contenant la liste des fichiers de votre arborescence
Remplacer le premier / de chaque ligne par un \$ dans le fichier *liste* et
 - a) enregistrer le résultat dans le fichier *new_liste*
 - b) enregistrer les lignes modifiées dans le fichier *modif_liste*
 - c) faites les deux à la fois
5. Donnez la commande *sed* permettant de supprimer les commentaires (lignes commençant par un #) et les lignes vides d'un fichier
6. Donner la ligne de commande qui affiche la liste des répertoires de connexion pour les utilisateurs définis dans */etc/passwd* sous la forme

 <lt/home/user>> est le répertoire de connexion de <ltuser>>
 <lt/home/user1>> est le répertoire de connexion de <ltuser1>>
7. Donner les lignes de commandes qui affichent pour le fichier *f*
 - a) les lignes de plus de 10 caractères
 - b) les lignes de moins de 5 caractères
 - c) les lignes de moins de 5 caractères et de plus de 100 caractères
8. Donner la ligne de commande qui pour le fichier *f*
 encadre la deuxième occurrence de nombre par des chevrons : <xxx>
9. Donner la ligne de commande qui pour le fichier *f*
 remplace chaque occurrence de chemin */.../.../...* par *http://.../.../...*
10. Donner la ligne de commande qui supprime les balises d'un document HTML
11. Donner la ligne de commande qui affiche le troisième caractère de chaque ligne
12. Donner la ligne de commande qui affiche du troisième au septième caractère de chaque ligne
13. Donner la ligne de commande qui affiche après le septième caractère de chaque ligne
14. Donner la ligne de commande qui affiche le contenu des fichiers de votre arborescence dont le nom commence et fini par un t, en remplaçant les chiffres par des dièses (#)