

ISS - Initiation aux Systèmes d'exploitation et au Shell

LU2IN020

TD 06 – Édition automatisée

Julien Sopena

octobre 2022

Le but de cette sixième semaine est d'étudier un exemple de commande permettant d'automatiser l'édition d'un flux. Comme le **grep**, ces commandes peuvent utiliser les expressions régulières étendues. Mais celles-ci s'en servent aussi pour formaliser des modifications à appliquer sur le contenu d'un flux de caractères. Les plus connus de ces commandes sont **awk** et **sed**. Nous avons ici choisi d'étudier cette dernière.

La commande **sed** qui signifie *Stream EDitor* est souvent vue comme un éditeur non interactif. En effet, elle permet d'automatiser la modification de contenu. Elle applique successivement l'algorithme suivant sur toutes les lignes lues sur son flux d'entrée ou dans le(s) fichier(s) passé(s) en paramètre :

1. remplissage du buffer par le contenu de la ligne lue ;
2. application des commandes sur le buffer ;
3. affichage sur la sortie standard du buffer (sauf si option **-n**).

Il existe de nombreuses commandes les plus courantes sont :

s/motif1/motif2/x : substitue dans le buffer le **motif2** au **x^e** (1^{er} par défaut) **motif1** ;

s/motif1/motif2/g : substitue dans le buffer le **motif2** à tous les **motif1** ;

y/list/list/ : transpose toutes les occurrences d'un caractère de la première liste par le caractère correspondant dans la deuxième liste, comme le fait la commande **tr** ;

d : efface le buffer ;

p : affiche le buffer ;

itexte : ajoute le texte avant le buffer ;

atexte : ajoute le texte après le buffer ;

Chaque commande peut être précédée d'un sélecteur d'adresses :

x ou **x,y** : la ligne doit être la **x^e** ou appartenir à l'intervalle **[x,y]**, où **x** et **y** sont des entiers ou un **\$** pour faire référence à la dernière ligne.

/motif/ : le buffer doit contenir le motif pour la commande s'exécute ;

! : inverse la sélection faite avec l'un des deux sélecteurs précédents.

Exercice 1 : Quelques équivalences

Question 1

Que font les commandes suivantes ?

```
sed '1d' mon_fichier.txt
sed 's/toto/titi/g' mon_fichier.txt
sed -E 's/toto/titi/g' mon_fichier.txt
sed -E '4s/toto/titi/2' mon_fichier.txt
sed -E '/toto/!d ; atutu' mon_fichier.txt
sed -E -n 's/toto/titi/' mon_fichier.txt
sed -E -n '10s/toto/titi/g ; 5,$p' mon_fichier.txt
sed -E -n '/tutu/s/toto/titi/p' mon_fichier.txt
```

Question 2

Donnez un équivalent `sed` de la commande `head -n 4 mon_fichier.txt` qui affiche les 4 premières lignes du fichier.

Question 3

Donnez un équivalent `sed` de la commande `tail -n +4 mon_fichier.txt` qui affiche tout le fichier à partir de la 4^e ligne.

Question 4

Donnez un équivalent `sed` de la commande `grep -E /expr/ mon_fichier.txt`.

Question 5

Donnez un équivalent `sed` de la commande `tr 'abc' 'xyz' mon_fichier.txt`.

Question 6

Beaucoup plus difficile (à faire à la fin du TD), donnez un équivalent `sed` de la commande `cut -f 3`.

Exercice 2 : Correcteur automatique

Question 1

Sachant que l'option `i` permet remplacer le contenu des fichiers passés en paramètre par le résultat de la commande, proposez une commande qui transforme les `#` en `//` dans un fichier `implem.c`.

Question 2

Donnez une commande qui permet d'effacer toutes les lignes vides ou ne contenant que des espaces.

Question 3

Donnez une commande qui permet d'ajouter une ligne vide après chaque ligne non vide.

Question 4

Donnez une commande qui permet d'ajouter un commentaire `### FONCTION ###` sur la ligne précédant la déclaration d'une fonction *Bash*.

Question 5

Donnez une commande qui permette d'ajouter un espace avant un `:`, un `?` ou un `!` s'il n'y en pas déjà un.

Question 6

Quelle commande permet de supprimer une occurrence d'un mot, lorsque celui-ci a été dupliqué par erreur? Vous commencerez par traiter le cas où l'on trouve deux fois le même mot consécutivement, avant d'étendre votre commande au cas où l'on trouve un nombre quelconque de copies. Pour simplifier, on supposera les copies sur la même ligne et séparées par un ou plusieurs espaces.

Question 7

Dans le texte d'une preuve, on décide de permuter les noms des variables *a* et *b* lorsqu'il y a un *c* sur la même ligne. Quelle commande permet de réaliser cette correction?