

ISS - Initiation aux Systèmes d'exploitation et au Shell

LU2IN020

TD 05 – Les expressions régulières

Julien Sopena

octobre 2022

Le but de cette cinquième semaine est d'étudier les expressions régulières, tant d'un point de vue syntaxique, que pratique. Ainsi, nous verrons leur mise-en-oeuvre dans le cadre de la commande `grep`.

Une expression régulière est une chaîne de caractères permettant de décrire un motif. Dans ce TD, comme dans le TP qui va suivre, nous allons utiliser les expressions régulières étendues POSIX. Notons qu'il en existe d'autres (vim, perl, python, java, ...) et que nous verrons qu'une partie de la norme.

Les expressions régulières sont composées de caractères que l'on peut préciser :

`c` : le caractère `c`, où `c` peut-être n'importe quel caractère *UTF-8* ;

`.` : un caractère quelconque et un seul ;

`[liste]` : un des caractères présents dans la liste ;

`[^liste]` : un caractère n'étant pas dans la liste.

Chaque caractère est considéré individuellement, mais un opérateur permet de les regrouper :

`(expr)` : toute l'expression est considérée comme un motif à part entière ;

`\x` : dite référence arrière `\x` est équivaut au `xe` groupement, où `x` est un entier.

Plusieurs opérateurs permettent de préciser la cardinalité d'une partie du motif :

`expr?` : ce qui précède est présent zéro ou une fois ;

`expr+` : ce qui précède est répété une ou plusieurs fois ;

`expr*` : ce qui précède est répété zéro, une ou plusieurs fois.

`expr{n}` : ce qui précède est répété exactement `n` fois ;

`expr{n,m}` : ce qui précède est répété exactement entre `n` fois et `m` fois ;

`expr{n,}` : ce qui précède est répété au moins `n` fois.

On peut composer les expressions :

`expr1 expr2` : concaténation des expressions `expr1` et `expr2` ;

`expr1|expr2` : l'une ou l'autre des expressions.

L'ensemble des derniers opérateurs permettent de définir un motif sans préciser sa position. On peut y ajouter deux derniers opérateurs pour préciser celle-ci :

`^` : ne correspond pas à un caractère, mais que ce doit être au début d'une ligne ;

`$` : ne correspond pas à un caractère, mais que ce doit être à la fin de la ligne.

Exercice 1 : Grep et scrabble

La commande `grep` signifie *Get REgular exPression*; elle permet de trouver une expression régulière dans un fichier. Dans toute la suite, nous l'utiliserons avec l'option `-E` qui indique à `grep` de utiliser que le dialecte des expressions régulières étendues (présenté plus haut).

Dans cet exercice on suppose l'accès à un fichier `dico.txt` contenant un dictionnaire français avec un mot en majuscule par ligne sans accent. Une version complète du dictionnaire est disponible à l'url <http://julien.sopena.fr/dico.txt>, ainsi qu'une version plus petite pour accélérer les tests http://julien.sopena.fr/dico_light.txt.

Question 1

Sur la grille, il y a déjà le mot `CHAT` et on se demande si l'on peut le compléter. Quelle commande permet d'afficher tous les mots contenant le mot `CHAT`.

Question 2

Sachant que l'option `-q` (synonyme de `--quiet`) permet de pas afficher les résultats, la commande se contentant de retourner 0 dès qu'elle trouve une occurrence, donnez une commande qui affiche "**"OK"** si le mot `TCHAT` fait partie du dictionnaire.

Question 3

Au scrabble on a 7 lettres en main. Quelle commande permet de compter tous les mots de 7 lettres ?

Question 4

Faites de même pour calculer combien de mots sont formés de 7 lettres ou moins ?

Question 5

Dans notre main, nous avons 3 fois la lettre `N`. Combien de mots de 7 lettres contiennent exactement 3 fois la lettre `N` ?

Question 6

Sur la table il y a déjà le mot `CHAT`. Avec notre main comprenant 3 `N`, on est peut-être mal parti. Quelle commande permet de trouver une façon de poser nos 7 lettres en complétant le mot `CHAT`? Pour simplifier, on supposera ici avoir de la chance pour les 4 autres lettres.

Question 7

Comme on vient de le voir, au scrabble, on utilise les lettres de sa main, mais aussi celles déjà posées sur la table. Comme la grille est un carré de 15 par 15, il est possible de combiner toutes ses lettres avec un mot de 8 lettres déjà posées sur la table. Afficher tous les mots de 15 lettres qui peuvent ainsi être formés ?

Question 8

Combien de mots a-t-on trouvés à la question précédente ?

Question 9

Un second tirage nous donne que des voyelles. On s'interroge donc sur leur distribution dans les mots français. Pour commencer, compter à l'aide d'une commande le nombre de mots ayant au moins deux voyelles de suite.

Question 10

Même question, mais avec deux fois la même voyelle.

Question 11

Proposez une commande permettant d'afficher tous les mots composés exclusivement d'une syllabe de deux lettres (une consonne et une voyelle) répétée deux fois.

Question 12

Pour finir de jouer avec les mots, proposez une commande qui cherche les palindromes (mot symétrique) de 7 lettres.

Exercice 2 : Mise en forme

La commande `tr` qui signifie *translate* permet de transposer et/ou d'éliminer des caractères lus sur son flux d'entrée standard :

`tr 'abc' 'xyz'` : transpose toutes les occurrences d'un caractère de la première liste par le caractère correspondant dans la deuxième liste (ici a par x, b par y et c par z) ;

`tr -s 'abc' 'xyz'` : idem, mais fusionne toutes les occurrences consécutives identiques. Ainsi ici, une suite bbb sera remplacée par un unique y, alors qu'en absence de l'option il y aurait eu trois y ;

`tr -d 'abc'` : supprime dans le flux tous les caractères de la liste.

À noter, que les listes peuvent contenir des intervalles, *i.e.*, '`d-g`' pour les lettres d, e, f et g.

Question 1

Quelle commande permet de transformer les espaces en `_` dans le fichier `cr.txt` ?

Question 2

Quelle commande permet de supprimer dans le flux d'entrée standard les accents du français sur les minuscules ?

Question 3

Quelle commande permet de transformer toutes les minuscules (non accentuées) en majuscules ?

Question 4

De nombreuses commandes, comme la commande `last` utilisée la semaine dernière, structurent leur affichage en colonne avec des espaces pour les rendre plus lisibles. Comment utiliser `tr` pour les transformer en format `csv`, où chaque colonne est séparée par des virgules ?

Question 5

Peut-on utiliser la commande `tr` pour corriger un format de commentaire `Bash #` en `//` format des commentaires en `C++` ?

Question 6

Dans le texte d'une preuve, on décide de permutez les noms des variables `x` et `y`. Pour ce faire, on propose d'exécuter le code suivant :



```
cat preuve.txt | tr 'x' 'y' | tr 'y' 'x'
```

Quel problème pose cette commande ? Proposez une solution.