1. Les expressions régulières

1.1. Présentation

Une expression régulière sert à identifier une chaîne de caractères répondant à un ou plusieurs critères comme par exemple une chaîne de caractères ne contenant des lettres.

Les expressions régulières permettent :

- de rechercher des “motifs” dans des fichiers texte comme les fichiers journaux ou les fichiers de configuration ;
- de réaliser, en une seule commande, un ensemble de tâches.
- appliquer des transformations dans du texte ;
- valider la conformité d’une donnée.

1.2. Définir des expressions régulières

Les expressions régulières utilisent divers opérateurs pour combiner des expressions de plus petite taille.

Il existe deux standards principaux en matière d'expression régulière, POSIX et Perl.

Un site pour apprendre à utiliser les expressions régulières :

- https://zestedesavoir.com/tutoriels/3651/les-expressions-regulieres-1/
- en testant les expressions régulières avec Rubular

Dans ce tutoriel vous utiliserez également :

- grep
- sed
- less

1.3. les commandes linux utiles

- **cd** : aller dans un dossier (absolu « /home/nouvel/test... » ou relatif « test/ »),
- **ls** : afficher les fichiers du dossier courant,
- **pwd** : afifcher le chemin du dossier courant,
- **mkdir** : créer un dossier,
- **rm** : effacer un fichier (ou un dossier : « rm -r »),
- **cat** : envoie le contenu d'un (des) fichier(s) vers le flux de sortie standard,
- **less** : visualiser un flux (fèches pour monter / descendre, « q » pour quitter),
- **grep** : afficher les lignes du fux qui correspondent à une expression régulière,
- **sort** : trie les lignes du flux par ordre lexicographique,
- **uniq** : omet les lignes répétées dans le fux,
- **wc** : compte le nombre de lignes / mots / caractères dans le flux.

1.4. Le pipe

Le pipe **|** permet de changer la destination d'un flux de données

1.4.1. Par exemple :

- Envoyer le contenu d'un fichier vers la sortie standard (terminal) :

```
cat fichier.txt
```

- Envoyer le contenu d'un fichier vers la visualisation paginée :

```
cat fichier.txt | less
```

- Envoyer le contenu d'un fichier vers **grep** pour sélectionne les lignes contenant le mot **limoges** et les le

</code> visualiser page par page :

```
cat fichier.txt | grep limoges | less
```

- Envoyer le contenu d'un fichier vers **grep**, qui sélectionne les lignes contenant le mot **limoges** puis enregistre les lignes dans un autre fichier :

```
cat fichier.txt | grep limoges > fichierlimoges.txt
```

1.4.1.1. Conseils

- tester des commandes dans le terminal,
- lorsque la sortie est volumineuse, ajoutez **| less** pour visualiser page par page ,
- pour enregistrer la sortie dans un fichier, ajoutez **> fchier.txt** ,
- pour voir (ou compter) des entrées distinctes, ajoutez **| sort | uniq -c**.

1.5. grep

grep signifie **Global Regular Expression Print** et est un outil en ligne de commande Linux / Unix utilisé pour rechercher une chaîne de caractères dans un fichier spécifié.

Le modèle de recherche de texte est appelé une **expression régulière (regex)**. Lorsqu'il trouve une correspondance, il imprime la ligne avec le résultat. La commande grep est pratique lors de la recherche dans de gros fichiers journaux.

grep peut s'utiliser directement pour rechercher un motif dans un fichier mais également via un **pipe**.

Les commandes suivantes alors équivalentes :

```
grep motif nomfichier
```

ou

```
cat nomfichier | grep motif
```

Pour rendre **grep** insensible à la casse, utilisez l'option -i :

Options utiles

option	Action	
-i	insensible à la casse	
-w	rechercher dans tous le fichiers	
-n	afficher les numéros de ligne	
-v	inverser la recherche pour exclure	
-mx	Limiter le nombre de sortie à x ligne	
-e chaine1 -e chaine2	Rechercher plusieurs chaînes	
-Ee “chaine1|chaine2”	Rechercher plusieurs chaînes avec les regex	
-c	compter le nombre de ligne	
“	**chaine”**	Rechercher en début de ligne
“chaine$”	Rechercher en fin de ligne	

Lien :

- https://man7.org/linux/man-pages/man1/grep.1.html
- https://www.malekal.com/exemples-de-commande-grep-sous-linux/

Découvrir d'autres usages utiles des expressions régulières avec grep sous Linux :

- https://www.malekal.com/exemples-de-commande-grep-sous-linux/

1.5.1. Liste de quelques méta-caractères utilisés et leur fonction :

Méta-caractère	Fonction
\d	Tous les chiffres
\D	Tout sauf les chiffres
.	Tout caractère
\w	Toutes les lettres
\W	Tout sauf des lettres

Méta-caractère	Fonction
[a-z]	Les lettres de a à z (attention les expressions régulières sont sensibles à la casse)
[0-9]	Les chiffres de 0 à 9
[abc]	a, b ou c
[^abc]	Ni a ni b ni c
{n}	Répéter n fois
{n,m}	Répéter de n à m fois
*	Répéter de 0 à n fois
+	Répéter de 1 à n fois
\s	Le caractère espace
\S	Tous les caractères sauf l'espace
Début de ligne ou négation	
$	Fin
\b	Début ou fin de mot
\	Caractère échappatoire
(...)	Sélection d'un groupe
|	Permettant de faire ou entre 2 sous-motifs

1.5.2. Quelques exemples d'expressions régulières et leur signification

Expression	Signification
test	Recherche de mots contenant test (tester, attester...)
\d{3}-\d{6}	Chaine de caractères de la forme 3chiffres-6chiffres comme 123-654321
a	Toutes les lignes commençant par la lettre a
\bf\w*\b	Tous les mots commençant par la lettre f
\d+	Tous les mots ne contenant que des chiffres (au moins un chiffre)
\w+	Tous les mots ne contenant que des lettres (au moins une lettre)
(fichier.+)\.pdf$	Va afficher toutes les lignes commençant par fichier avec l'extension .pdf mais ne va afficher que la partie avant l'extension (fichier1.pdf affichera fichier1)
ABC|XYZ	Tous les mots contenant ABC ou XYZ
\btest\b	Tous les mots « test » exactement (on ne cherche que le mot test)
(\d+)x(\d+)	Pour le texte 1920×800 va afficher deux groupes 1920 et 800

From:
https://siocours.lycees.nouvelle-aquitaine.pro/ - **Les cours du BTS SIO**

Permanent link:
https://siocours.lycees.nouvelle-aquitaine.pro/doku.php/reseau/regexp/accueil?rev=1638278504

Last update: **2021/11/30 13:21**