Systèmes UNIX. Expressions régulières

Sergey Kirgizov

Aujourd'hui

Expressions régulières, commandes grep et sed.

Grep

grep : recherche de motifs dans un texte

commande `grep` : filtrage de lignes suivant un motif

- Permet de filtrer les lignes d'un fichier et de n'afficher que certaines d'entre elles
- Filtrage par mot, expressions, motif, ...
- Utile pour savoir si un fichier contient un mot donné
- Nombreuses applications pour les fichiers de configuration
- Usage : `grep` *`motifRecherché fichier(s)`*
  - option `-v` : afficher les lignes qui ne contiennent pas le motif
  - option `-n` : numéroter les lignes résultat
  - sans fichier en paramètre : utilisation de l'entrée standard
  - option `-e` : pour préciser plusieurs motifs
  - option `-E` : utiliser les expressions régulières étendues
  - option `-o` : afficher seulement le motif, pas toute la ligne

Exemple d'utilisation de grep (filtres simples)

Affichage du fichier texte de référence :

```
1  Galactica:fichiers benoit$ cat introStarWarsFull.txt
2  Episode IV - Un nouvel espoir
3
4  C'est une période de guerre civile. Des vaisseaux
5  spatiaux rebelles, frappant à partir d'une base cachée,
6  ont remporté leur première victoire contre l'empire
7  galactique maléfique.
8
9  Durant la bataille, les espions rebelles ont réussi
10 à dérober les plans secrets de l'arme ultime de l'Empire,
11 l'Etoile de la Mort - une station spatiale fortifiée
12 avec suffisamment de puissance de feu pour détruire une
13 planète entière...
14
15 Poursuivie par les sinistres agents de l'Empire, la
16 princesse Leia rentre chez elle en hâte à bord de son
17 vaisseau stellaire, gardienne des plans volés qui
18 pourraient sauver son peuple et rétablir la liberté
19 dans la galaxie...
```

Exemple d'utilisation de grep (filtres simples)

Afficher les lignes qui contiennent le mot "rebelles"

```
1  Galactica:fichiers benoit$ grep 'rebelles' introStarWarsFull.txt
2  spatiaux rebelles, frappant à partir d'une base cachée,
3  Durant la bataille, les espions rebelles ont réussi
```

Afficher les lignes qui ne contiennent pas la suite de lettres 'le'

```
1  Galactica:fichiers benoit$ grep -v 'le' introStarWarsFull.txt
2  Episode IV - Un nouvel espoir
3
4  galactique maléfique.
5
6  avec suffisamment de puissance de feu pour détruire une
7  planète entière...
8
9  vaisseau stellaire, gardienne des plans volés qui
10 dans la galaxie...
```

Afficher depuis l'entrée standard toutes les lignes qui contiennent soit le mot 'Empire', soit le mot 'vaisseau'

```
1  Galactica:fichiers benoit$ cat introStarWarsFull.txt | grep -e 'Empire' -e 'vaisseau'
2  C'est une période de guerre civile. Des vaisseaux
3  à dérober les plans secrets de l'arme ultime de l'Empire,
4  Poursuivie par les sinistres agents de l'Empire, la
5  vaisseau stellaire, gardienne des plans volés qui
```

Exemple d'utilisation de grep (filtres simples)

Question

Comment n'afficher que les lignes qui contiennent à la fois les mots 'plans' et 'Empire' ?

Réponse :

- On sélectionne d'abord les lignes qui contiennent le mot 'plan'
- Parmi ces dernières, on sélectionne celles qui contiennent le mot 'Empire', en utilisant les tubes
- `grep 'plans' introStarWarsFull.txt | grep 'Empire'`

Application :

```
1 Galactica:fichiers benoit$ grep 'plans' introStarWarsFull.txt  | grep 'Empire'
2 à dérober les plans secrets de l'arme ultime de l'Empire,
```

Exemple d'utilisation de grep (filtres simples)

Question

Comment n'afficher que les lignes qui contiennent à la fois les mots 'plans' et 'Empire' ?

Réponse :

- On sélectionne d'abord les lignes qui contiennent le mot 'plan'
- Parmi ces dernières, on sélectionne celles qui contiennent le mot 'Empire', en utilisant les tubes
- `grep 'plans' introStarWarsFull.txt | grep 'Empire'`

Application :

```
1 Galactica:fichiers benoit$ grep 'plans' introStarWarsFull.txt  | grep 'Empire'
2 à dérober les plans secrets de l'arme ultime de l'Empire,
```

Expressions régulières

Limites des filtres simples

- Faciles à utiliser mais sont très limités
- Filtrage basique réalisé sur une suite exacte de caractères
- Pour palier à ces limitations, utilisation des expressions régulières

Expressions régulières (ou expressions rationelles)

- Motif qui décrit un ensemble de chaines de caractères possibles
- Extrêmement utilisés en informatique, sous UNIX
- Exception : pas utilisé dans les lignes de commandes shell
- Beaucoup plus compliqué à utiliser que les filtres simples
- Mais beaucoup plus puissant

Expressions régulières : principes

Principe d'utilisation des expressions régulières

- Définition de classes de caractères
- Utilisation de quantificateurs (caractères ?, +, *, {..}) sur ces classes de caractères
- Permet d'imposer qu'un motif donné soit répété une ou plusieurs fois
- Possibilité d'utiliser l'opérateur 'ou' (caractère '|')
- Application des opérateurs et quantificateurs sur un caractère , un groupe de caractères `(...)` ou un ensemble `[...]`
- Utilisation de délimiteurs de débuts et de fin de ligne
- A utiliser avec l'outil `grep` et option `-E`, ou avec `egrep`

Syntaxe des expressions régulières (1/2)

Syntaxe

- Délimiteurs de chaînes :

^	Définit le début de la ligne
$	Définit la fin de la ligne

- Motifs simples :

c	Recherche du caractère *c*
cde	Recherche de la suite de caractères *cde*
.	Joker pour remplacer exactement un caractère
[. . .]	Domaine de caractères autorisés : un caractère de cet ensemble est requis
[a-z]	ensemble des lettres minuscules
[a-f0-9]	lettres minuscules de A à F, et des chiffres
[^ . . .]	Inverse du domaine de caractères : un caractère qui n'appartient pas à ce domaine est requis

Afficher les lignes qui contiennent un un m minuscule :

```
Galactica:fichiers benoit$ grep -E 'm'  introStarWarsFull.txt
```

Afficher les lignes qui commencent par un A majuscule :

```
Galactica:fichiers benoit$ grep -E '^A'  introStarWarsFull.txt
```

Afficher les lignes qui contiennent 'plan', 'empire', ou 'rebelle' :

```
Galactica:fichiers benoit$  grep -E 'plan|empire|rebelle'   introStarWarsFull.txt
```

Afficher les lignes qui ne commencent pas par une majuscule :

```
Galactica:fichiers benoit$  grep -E '^[^A-Z]'   introStarWarsFull.txt
```

Afficher les lignes qui ne sont pas vides :

```
Galactica:fichiers benoit$ grep -v -E ^$  introStarWarsFull.txt
# ou encore
Galactica:fichiers benoit$ grep -E .  introStarWarsFull.txt
```

Afficher les lignes qui contiennent Le, le , La ou la :

```
Galactica:fichiers benoit$ grep -E '[Ll][ea] ' introStarWarsFull.txt
```

Syntaxe des expressions régulières (2/2)

Syntaxe (suite)

- Définition de clauses disjonctives :

$a \mid b$	Recherche du motif a ou alors du motif b

- Quantificateurs (appliqués sur le motif le précédant) :

*	0, 1 ou plusieurs fois ce motif
+	1 ou plusieurs fois ce motif
?	0 ou 1 fois ce motif
$\{n\}$	exactement n fois ce motif
$\{m,n\}$	entre m et n fois ce motif
$\{m,\}$	au moins m fois ce motif
(...)	Permet de grouper des caractères pour appliquer un quantificateur sur l'ensemble du groupe. Permet également d'affecter les mots trouvés dans des variables (utilisation avec sed)

Afficher les lignes contenant long, loong, looong, . . . :

```
Galactica:fichiers benoit$ grep -E 'lo+ng'  fichier.txt
```

Même chose, syntaxe différente

```
Galactica:fichiers benoit$ grep -E 'lo(o*)ng'  fichier.txt
```

Afficher les lignes contenant un mot de 4 caractères ou plus :

```
 Galactica:fichiers benoit$ grep -E '[A-Za-z0-9]{4,}' fichier.txt
```

Afficher les lignes contenant Le, le , La, la, un, une, Un ou Une :

```
Galactica:fichiers benoit$ grep -E '([Ll][ea])|([uU]ne?)' fichier.txt
```

Afficher les lignes contenant cinq ou six mots :

```
Galactica:fichiers benoit$ grep -E '^([A-Za-z0-9] ){4,5}([A-Za-z0-9])$' fichier.txt
#probleme avec les accents : la seconde ligne est mieux
Galactica:fichiers benoit$ grep -E '^([^ ]* ){4,5}[^ ]*$' fichier.txt
```

Afficher tous les mots d'exactement 3 caractères (plus difficile) :

```
Galactica:fichiers benoit$ grep -o -E '(^| )[A-Za-z0-9]{3}( |$)' fichier.txt
```

Expressions régulières : applications

Remarque

Pour utiliser un caractère spécial comme caractère standard, on le précède d'un antislash \

Question

Quels sont tous les mots reconnus par les expressions régulières suivantes :

1. `([1-9]|[0-2][1-9]|30|31)[-/\\](0?[1-9]|1[0-2])`
2. `[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}`

Réponse :

1. Dates d'anniversaires : 17/07 , 14/02, 4-5, 13/5, ...
2. E-mails

Backreferences

On utilise \x pour se référer à une partie de texte correspondant à une partie d'expression régulière comprise entre les parenthèses numéro x.

Exemple :

La regex

- (..)test(..A)\1\2

va correspondre à

- 55test-+A55-+A
- MYtestMyAMYMyA

mais pas à

- 55test-+A44-+A

Sed

sed - Stream EDitor

sed : éditeur de flux à la volée

- Outil extrêmement puissant de UNIX (mais compliqué)
- Permet des manipulations avancées sur un fichier texte
- Utilise des expressions régulières associées à des commandes

exemple de manipulations réalisables avec sed

- Passer certains mots du texte en majuscule seulement
- Remplacer un groupe de mots par un autre
- Changer des formats de date (17/07/81 en 17 Juillet 1981)
- Renommer un ensemble de fichiers de manière uniforme.
- Et bien d'autres

sed - Stream EDitor

Principe de fonctionnement : éditeur de flux à la volée

1. Besoin d'un flux d'entrée : données à modifier. Proviennent d'un fichier ou de l'entrée standard
2. Définition des opérations à réaliser avec syntaxe spécifique
3. Edition du résultat sur la sortie standard

syntaxe : `sed 'operation' [fichier]`

une opération très utilisée : le remplacement

1. opération : `'s/avant/après/flag'`
2. Force de l'outil : *avant* peut être une expression régulière (-E)
3. remplace la 1^{re} **occurence par ligne**, sauf si flag='g' ou num
4. Identification d'éléments dans *avant* réutilisables dans *après*
5. Edition du résultat sur la sortie standard

Réutilisation du fichier `introStarWarsFull.txt` :

```
1  Galactica:fichiers benoit$ cat introStarWarsFull.txt
```

```
1  Episode IV - Un nouvel espoir
2
3  C'est une période de guerre civile. Des vaisseaux
4  spatiaux rebelles, frappant à partir d'une base cachée,
5  ont remporté leur première victoire contre l'empire
6  galactique maléfique.
7
8  Durant la bataille, les espions rebelles ont réussi
9  à dérober les plans secrets de l'arme ultime de l'Empire,
10 l'Etoile de la Mort - une station spatiale fortifiée
11 avec suffisamment de puissance de feu pour détruire une
12 planète entière...
13
14 Poursuivie par les sinistres agents de l'Empire, la
15 princesse Leia rentre chez elle en hâte à bord de son
16 vaisseau stellaire, gardienne des plans volés qui
17 pourraient sauver son peuple et rétablir la liberté
18 dans la galaxie...
```

Remplacer 'rebelles' par 'de la rebellion' dans un texte

```
Galactica-2:fichiers benoit$ sed 's/rebelles/de la rebellion/g' introStarWarsFull.txt
```

```
Episode IV - Un nouvel espoir

C'est une période de guerre civile. Des vaisseaux
spatiaux de la rebellion, frappant à partir d'une base cachée,
ont remporté leur première victoire contre l'empire
galactique maléfique.

Durant la bataille, les espions de la rebellion ont réussi
à dérober les plans secrets de l'arme ultime de l'Empire,
l'Etoile de la Mort - une station spatiale fortifiée
avec suffisamment de puissance de feu pour détruire une
planète entière...

Poursuivie par les sinistres agents de l'Empire, la
princesse Leia rentre chez elle en hâte à bord de son
vaisseau stellaire, gardienne des plans volés qui
pourraient sauver son peuple et rétablir la liberté
dans la galaxie...
```

- note : fichier non modifié, seul le résultat apparaît à l'écran
- Peut on faire des choses que Word ne fait pas ?

Remplacer tous les mots de 5 ou 6 lettres par '_____'

```
1  sed -E "s/(^| )[A-Za-z]{5,6}( |$)/ _____ /g" introStarWarsFull.txt
```

```
1  Episode IV - Un _____ espoir
2
3  C'est une période de _____ civile. Des vaisseaux
4  spatiaux rebelles, frappant à _____ d'une base cachée,
5  ont remporté leur première victoire _____ l'empire
6  galactique maléfique.
7
8   _____ la bataille, les espions rebelles ont réussi
9  à dérober les _____ secrets de l'arme _____ de l'Empire,
10 l'Etoile de la Mort - une station spatiale fortifiée
11 avec suffisamment de puissance de feu pour détruire une
12 planète entière...
13
14 Poursuivie par les sinistres _____ de l'Empire, la
15 princesse Leia _____ chez elle en hâte à bord de son
16 vaisseau stellaire, gardienne des _____ volés qui
17 pourraient _____ son _____ et rétablir la liberté
18 dans la galaxie...
```

Supprimer toutes les voyelles :

```
1  sed -E "s/[aeiouAEIOUéèÃªàâ]//g" introStarWarsFull.txt
```

```
 1  psd V - n nvl spr
 2
 3  C'st n prd d grr cvl. Ds vssx
 4  sptx rblls, frppnt  prtr d'n bs cch,
 5  nt rmprt lr prmr vctr cntr l'mpr
 6  glctq mlfq.
 7
 8  Drnt l btll, ls spns rblls nt rss
 9   drbr ls plns scrts d l'rm ltm d l'mpr,
10  l'tl d l Mrt - n sttn sptl frtf
11  vc sffsmmnt d pssnc d f pr dtrr n
12  plnt ntr...
13
14  Prsv pr ls snstrs gnts d l'mpr, l
15  prncss L rntr chz ll n ht  brd d sn
16  vss stllr, grdnn ds plns vls q
17  prrnt svr sn ppl t rtblr l lbrt
18  dns l glx...
```

sed - Stream EDitor

Nommage et réutilisation d'éléments lors de substitution

1. Nommer un motif d'une expression régulière : `( motif )`
2. Note : réutilisation des parenthèse initialement utilisées pour choix entre motif
3. Réutilisation : \#, où # est le numéro du motif
4. & correspond à toute la chaine de l'expression régulière

Résumer le texte aux premiers et derniers mots de chaque ligne

```
1  sed -E "s/^([^ ]*) ([^ ]*) .* ([^ ]*) ([^ ]*)$/\1 \2 ... blabla ... \3 \4/"
       introStarWarsFull.txt
```

```
1  Episode IV ... blabla ... nouvel espoir
2
3  C'est une ... blabla ... Des vaisseaux
4  spatiaux rebelles, ... blabla ... base cachée,
5  ont remporté ... blabla ... contre l'empire
6  galactique maléfique.
7
8  Durant la ... blabla ... ont réussi
9  à dérober ... blabla ... de l'Empire,
10 l'Etoile de ... blabla ... spatiale fortifiée
11 avec suffisamment ... blabla ... détruire une
12 planète entière...
13
14 Poursuivie par ... blabla ... l'Empire, la
15 princesse Leia ... blabla ... de son
16 vaisseau stellaire, ... blabla ... volés qui
17 pourraient sauver ... blabla ... la liberté
18 dans la galaxie...
```

Questions

Écrire une expression régulière pour matcher les mots mots de l'alphabet unaire (contenant uniquement un seul symbole) :

- de longueur paire ;
- dont la longueur est divisible par 3 ;
- dont la longueur est un nombre composé.

Par exemple 0, 000, 00, 0000 sont les mots de l'alphabet unaire contenant uniquement 0.

Le mot 0000000000 a la longueur composé, $10 = 2 \cdot 5$.

Théorie

- Automate fini
- Grammaire régulière
- Expression régulière

`https://regexper.com`

Merci Benoît !

Ce cours est fait en partie à partir du cours de Benoît Darties.
`https://benoit.darties.fr/`