

Dernière mise-à-jour : 2020/01/30 03:27

# 103.7 - Rechercher dans des fichiers textes en utilisant des expressions régulières (2/60)

## Expressions Régulières

La manipulation de fichiers textes utilise des **expressions régulières**. Sous Linux il existe deux types d'expressions régulières :

- expressions régulières basiques - IEEE POSIX Basic Regular Expressions, appelées **ERb**,
  - utilisées par les commandes **vi**, **grep**, **expr** et **sed**,
- expressions régulières étendues - IEEE POSIX Extended Regular Expressions, appelées **ERe**,
  - utilisées par les commandes **egrep** ( grep -E ) et **awk**.

Les expressions régulières utilisent des caractères spéciaux. Certains caractères sont communs aux Erb et aux Ere :

Caractère spécial	Description
^	Trouver la chaîne au début de la ligne
\$	Trouver la chaîne à la fin de la ligne
\	Annuler l'effet spécial du caractère suivant
[ ]	Trouver n'importe quel des caractères entre les crochets
[^]	Exclure les caractères entre crochets
.	Trouver n'importe quel caractère sauf à la fin de la ligne
*	Trouver 0 ou plus du caractère qui précède
\<	Trouver la chaîne au début d'un mot
\>	Trouver la chaîne à la fin d'un mot

### ERb

Certains caractères spéciaux sont spécifiques aux ERb :

Caractère spécial	Description
\{x,y\}	Trouver de <b>x</b> à <b>y</b> occurrences de ce qui précède
\{x\}	Trouver exactement le nombre <b>x</b> d'occurrences de ce qui précède
\{x,\}	Trouver le nombre <b>x</b> ou plus d'occurrences de ce qui précède
\(ERb)	Mémoriser une ERb
\1	Rappeler la première ERb mémorisée
\2, \3 ...	Rappeler la deuxième ERb mémorisée, rappeler la troisième ERb mémorisée etc

## ERe

Certains caractères spéciaux sont spécifiques aux ERe :

Caractère spécial	Description
?	Trouver 0 ou 1 occurrence de ce qui précède
+	Trouver 1 ou <b>n</b> d'occurrences de ce qui précède
{x,y}	Trouver de <b>x</b> à <b>y</b> occurrences de ce qui précède
{x}	Trouver exactement le nombre <b>x</b> d'occurrences de ce qui précède
{x,}	Trouver le nombre <b>x</b> ou plus d'occurrences de ce qui précède
()	Faire un <b>ET</b> des expressions régulières entre les parenthèses
	Faire un <b>OU</b> des expressions régulières se trouvant de chaque côté du pipe

# Outils de Manipulation de Fichiers Texte

## La commande grep

La commande grep peut être utilisée pour rechercher des lignes contenant une chaîne de caractères dans un jeu de fichiers.

Par défaut, la commande grep est sensible à la casse. Pour rendre cette commande insensible à la casse, il faut utiliser l'option **-i**.

La commande grep peut être aussi utilisée pour faire l'inverse, autrement dit de montrer les lignes qui ne contiennent pas la chaîne recherchée. Dans ce cas, il faut utiliser l'option **-v**.

La commande grep peut être utilisée avec des **Expressions Régulières basiques**. Ceci est utile pour rechercher dans le contenu de fichiers.

### Options de la commande



**A faire :** Utilisez l'option **-help** de la commande **grep** pour visualiser les options de la commande.

### La Commande egrep

La commande **egrep** est identique à la commande **grep -E**. Dans les deux cas, l'utilisation des expressions régulières est étendue aux ERe.

### Options de la commande



**A faire :** Utilisez l'option **-help** de la commande **egrep** pour visualiser les options de la commande.

### La Commande fgrep

La commande **fgrep** est identique à la commande **grep -F**. Dans les deux cas et par défaut la recherche concerne une chaîne de caractères interprétés dans un sens littéral sans utilisation de caractères spéciaux ni d'expressions régulières.

## Options de la commande



**A faire :** Utilisez l'option **-help** de la commande **fgrep** pour visualiser les options de la commande.

### LAB #1 - Utiliser grep, egrep et fgrep

Téléchargez le fichier **greptest** vers votre machine virtuelle en double-cliquant sur le titre ci-dessous :

[greptest](#)

```
fenestrOS
fenestros
555-5555
f
.fenestros
.fe
£
```

Placez ce fichier dans le répertoire **/tmp** :

```
[root@centos7 ~]# mv /home/trainee/Downloads/greptest /tmp/greptest
```

Recherchez maintenant toute ligne du fichier **/tmp/greptest** contenant au moins une lettre :

```
[root@centos7 ~]# grep '[a-zA-Z]' /tmp/greptest
fenestrOS
fenestros
f
```

```
.fenestros  
.fe
```

Recherchez maintenant toute ligne contenant au moins une lettre ou un chiffre :

```
[root@centos7 ~]# grep '[a-zA-Z0-9]' /tmp/greptest  
fenestrOS  
fenestros  
555-5555  
f  
.fenestros  
.fe
```



**Important** : Notez la présence de la ligne 555-5555.

Recherchez maintenant toute ligne contenant un numéro de téléphone au format NNN-NNNN :

```
[root@centos7 ~]# grep '[0-9]\{3\}-[0-9]\{4\}' /tmp/greptest  
555-5555
```

Recherchez maintenant toute ligne contenant exactement un caractère :

```
[root@centos7 ~]# grep '^.$' /tmp/greptest  
f  
£
```



**Important** : Notez l'utilisation des caractères spéciaux le début de ligne : ^, n'importe quel caractère : . et la fin de ligne : \$.

Recherchez maintenant toute ligne commençant par un point :

```
[root@centos7 ~]# grep '^\. ' /tmp/greptest
.fenestros
.fe
```



**Important** : Notez l'utilisation du caractère d'échappement \ pour annuler l'effet du caractère spécial .



La commande grep peut aussi être utilisée pour rechercher une chaîne dans tous les fichiers d'un répertoire spécifié : **grep -rnw 'directory' -e "pattern"**. Vous pouvez aussi spécifier les extensions des fichiers dans lesquels vous voulez rechercher la chaîne : **grep -include={\*.doc,\*.xls} -rnw 'directory' -e "pattern"**. Dernièrement vous pouvez exclure des fichiers de la recherche de la façon suivante : **grep -exclude=\*.doc -rnw 'directory' -e "pattern"**.

Téléchargez le fichier **greptest** modifié vers votre machine virtuelle en double-cliquant sur le titre ci-dessous :

[greptest](#)

```
# Starting comment
fenestrOS
fenestros
# Another comment
555-5555
f

.fenestros

.fe
```

```
£  
# End comment
```

Placez ce fichier dans le répertoire **/tmp** :

```
[root@centos7 ~]# mv /home/trainee/Downloads/greptest /tmp/greptest
```

Utilisez maintenant la commande **grep** avec l'option **-E** pour supprimer les lignes de commentaires ainsi que les lignes vides :

```
[root@centos7 ~]# grep -E -v '^(#|$)' /tmp/greptest  
fenestrOS  
fenestros  
555-5555  
f  
.fenestros  
.fe  
£
```



**Important** : Notez l'utilisation des parenthèses pour faire un regroupement ainsi que le pipe pour représenter un OU. L'expression '^(#|\$)' indique donc "toute ligne commençant par le caractère #" OU "toute ligne où le début de la ligne est aussi la fin de la ligne".

Utilisez maintenant la commande **egrep** pour envoyer le contenu du fichier **/tmp/greptest**, sans commentaires et sans lignes vides, dans le fichier **/tmp/greptest1** :

```
[root@centos7 ~]# egrep -v '^(#|$)' /tmp/greptest > /tmp/greptest1  
[root@centos7 ~]# cat /tmp/greptest1  
fenestrOS  
fenestros  
555-5555  
f
```

```
.fenistros  
.fe  
£
```



**Important :** Cette commande est particulièrement utile face à un fichier de configuration de plusieurs centaines de lignes dont certaines contiennent des directives activées d'autres sont vides ou en commentaires. De cette façon vous pouvez générer facilement un fichier ne contenant que les directives activées.

Téléchargez le fichier **greptest** modifié vers votre machine virtuelle en double-cliquant sur le titre ci-dessous :

[greptest](#)

```
# Starting comment  
^ This line will be used to demonstrate the use of fgrep  
fenestrOS  
fenistros  
# Another comment  
555-5555  
f  
  
.fenistros  
  
.fe  
  
£  
# End comment
```

Placez ce fichier dans le répertoire **/tmp** :

```
[root@centos7 ~]# mv /home/trainee/Downloads/greptest /tmp/greptest
```

Utilisez maintenant la commande **fgrep** pour rechercher la ligne commençant par le caractère ^ :

```
[root@centos7 ~]# fgrep '^' /tmp/greptest
^ voici une ligne pour la recherche fgrep
```

Comparez le résultat ci-dessus avec celui de la commande grep :

```
[root@centos7 ~]# grep '^' /tmp/greptest
# Commentaire du début
^ voici une ligne pour la recherche fgrep
fenestrOS
fenestros
# Un autre commentaire
555-5555
f

.fenestros

.fe

£
# Commentaire de la fin
```

En effet, la ligne de commande en utilisant la commande grep devrait être :

```
[root@centos7 ~]# grep '^'^ /tmp/greptest
^ voici une ligne pour la recherche fgrep
```

## Le Commande sed

La commande **sed** ou *Stream EDitor* est un éditeur de texte non-interactif. Les actions spécifiées par la commande sed sont exécutées par défaut sur chaque ligne du fichier. La commande sed ne modifie pas le fichier d'origine et sa sortie standard est le canal 1.

Si plusieurs actions sont spécifiées dans la ligne de commande, chacune doit être précédée par l'option **-e**.

La syntaxe de la commande sed est la suivante :

```
sed [adresse] commande [arguments]
```

L'**adresse** permet de stipuler les lignes concernées par la **commande**.

La syntaxe d'une adresse peut être :

adresse	Lignes concernées
a	La ligne numéro <b>a</b>
\$	La dernière ligne
/ERb/	Les lignes qui correspondent à l'ERb
a,b	De la ligne numéro <b>a</b> jusqu'à la ligne numéro <b>b</b>
/ERb1/, /ERb2/	Toutes les lignes entre la première occurrence correspondant à l'ERb1 jusqu'à la première occurrence correspondant à l'ERb2

Le commandes de sed sont :

commande	Description
d	Ne pas afficher la ou les ligne(s)
p	Afficher la ou les ligne(s)
s	Effectuer une substitution
w	Ecrire le ou les ligne(s) dans un fichier
=	Afficher le numéro de la ligne spécifiée
!	Exécuter la commande ci-dessus sur toutes les lignes sauf celle spécifiées dans l'adresse

## Options de la commande



**A faire** : Utilisez l'option **-help** de la commande **sed** pour visualiser les options de la commande.

## LAB #2 - Utiliser la Commande sed

La commande **d** de sed permet de ne pas afficher certaines lignes à l'écran. Dans l'exemple qui suit, les 10 premières lignes du fichier **/etc/services** ne sont pas affichées à l'écran :

```
[root@centos7 ~]# sed '1,10d' /etc/services | more
# are included, only the more common ones.
#
# The latest IANA port assignments can be gotten from
#      http://www.iana.org/assignments/port-numbers
# The Well Known Ports are those from 0 through 1023.
# The Registered Ports are those from 1024 through 49151
# The Dynamic and/or Private Ports are those from 49152 through 65535
#
# Each line describes one service, and is of the form:
#
# service-name    port/protocol  [aliases ...]  [# comment]

tcpmux          1/tcp           # TCP port service multiplexer
tcpmux          1/udp           # TCP port service multiplexer
rje             5/tcp           # Remote Job Entry
rje             5/udp           # Remote Job Entry
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
systat          11/udp          users
daytime         13/tcp
--Plus--
```

Dans l'exemple qui suit, sed n'affiche pas de lignes de commentaires, c'est-à-dire les lignes commençant par le caractère **#** :

```
[root@centos7 ~]# sed '/^#/d' /etc/services | more

tcpmux          1/tcp           # TCP port service multiplexer
tcpmux          1/udp           # TCP port service multiplexer
rje             5/tcp           # Remote Job Entry
rje             5/udp           # Remote Job Entry
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
systat          11/udp          users
daytime         13/tcp
daytime         13/udp
qotd            17/tcp           quote
qotd            17/udp           quote
msp              18/tcp           # message send protocol
msp              18/udp           # message send protocol
chargen         19/tcp           ttystt source
chargen         19/udp           ttystt source
ftp-data        20/tcp
ftp-data        20/udp
ftp              21/tcp
ftp              21/udp           fspd fspd
--Plus--
```



**Important :** Notez que l'ERb est entourée des caractères / et /.

La commande sed vous permet d'afficher à l'écran certaines lignes spécifiées en utilisant la commande **p** :

```
[root@centos7 ~]# sed '1,2p' /etc/passwd
```

```
root:x:0:0:root:/bin/bash
root:x:0:0:root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
...
...
```



**Important :** Notez que sed affiche également tout le contenu du fichier. Ceci implique que les lignes 1 et 2 s'affichent deux fois.

Pour n'afficher que les lignes spécifiées, il convient d'utiliser l'option **-n** :

```
[root@centos7 ~]# sed -n '1,2p' /etc/passwd
root:x:0:0:root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
```

La commande **w** permet d'écrire dans un fichier. Par exemple pour écrire dans le fichier **/tmp/sedtest** toutes les lignes du fichier **/etc/services** ne commençant pas par le caractère **#**, il convient d'utiliser la commande suivante :

```
[root@centos7 ~]# sed -n '/^#/!w /tmp/sedtest' /etc/services
[root@centos7 ~]# more /tmp/sedtest

tcpmux      1/tcp          # TCP port service multiplexer
tcpmux      1/udp          # TCP port service multiplexer
rje         5/tcp          # Remote Job Entry
rje         5/udp          # Remote Job Entry
echo        7/tcp
echo        7/udp
discard     9/tcp          sink null
discard     9/udp          sink null
```

```
systat      11/tcp      users
systat      11/udp      users
daytime     13/tcp
daytime     13/udp
qotd        17/tcp      quote
qotd        17/udp      quote
msp          18/tcp      # message send protocol
msp          18/udp      # message send protocol
chargen     19/tcp      ttystt source
chargen     19/udp      ttystt source
ftp-data    20/tcp
ftp-data    20/udp
ftp          21/tcp
ftp          21/udp      fspd fspd
--Plus-- (0%)
```

La commande **s** permet de procéder à une substitution :

```
[root@centos7 ~]# echo "user1,user2,user3" > /tmp/sedtest1
[root@centos7 ~]# cat /tmp/sedtest1 | sed 's/,/ /g'
user1 user2 user3
```



**Important** : Notez que dans cet exemple, la commande s est suivi par un argument qui prend la forme /ce qui est à remplacer (caractère, chaîne ou ERb)/chaîne de remplacement/g. Le caractère g force le remplacement de toutes les occurrences. Sans elle, uniquement la première occurrence serait remplacée. Dans le cas de l'exemple, on remplace donc les virgules par des espaces.

<html>

Copyright © 2004-2017 Hugh Norris.<br><br> <a rel="license" href="http://creativecommons.org/licenses/by-nc-nd/3.0/fr/"></a><br />Ce(tte) oeuvre est mise à disposition selon les termes de la <a rel="license" href="http://creativecommons.org/licenses/by-nc-nd/3.0/fr/">Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France</a>.

</html>

---

From:

<https://www.ittraining.team/> - **www.ittraining.team**



Permanent link:

<https://www.ittraining.team/doku.php?id=elearning:workbooks:french:14:junior:l116>

Last update: **2020/01/30 03:27**