grep

- Objet : filtre grep
- Niveau requis :
  débutant, avisé
- Commentaires : *Recherche de caractères dans des fichiers textes.*
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !
- Suivi :
  - Création par smolski le 17/01/2009
  - Testé par smolski le 18/10/2012
- Commentaires sur le forum : c'est ici [1)]

Introduction

L'acronyme[2)] grep vient de la contraction de **Get Regular Expression Print**, *Print* signifiant l'affichage.

Le programme grep explore un ou une série de fichiers d'un ou plusieurs répertoires à la recherche de textes filtrés par une expression régulière donnée (*des lettres, des chiffres*...).

TP01

Créer les répertoires et fichiers exemples

Créez un répertoire test1 avec un fichier nommé essai.txt comportant des noms et prénoms avec la commande cat :

```
mkdir test1
```

Créez le fichier[3)] essai.txt ainsi :

```
touch essai.txt
```

Et rédigez[4)] ce fichier **test1/essai.txt** ainsi :

```
cat > test1/essai.txt <<EOF
tartempion eric
greg lucien
howard charles
santiago germaine
EOF
```

Utilisez la commande cat pour en vérifier le contenu :

```
cat test1/essai.txt
```

retour de la commande

```
tartempion eric
greg lucien
howard charles
santiago germaine
```

De la même façon que précédemment, créons un second répertoire nommé `test2` dans lequel nous créons le fichier **adresse.c** contenant les chaînes de caractères suivantes :

```
cat > test2/adresse.c <<EOF
25, rue de la source
5bis, avenue linux
3358, street gandhi
EOF
```

On vérifie les deux fichiers d'un coup ainsi :

```
cat ~/test1/essai.txt ~/test2/adresse.c
```

retour de la commande

```
tartempion eric
greg lucien
howard charles
santiago germaine
25, rue de la source
5bis, avenue linux
3358, street gandhi
```

Exemples Pratiques

Si l'on omet le nom de fichier à traiter, `grep` agit sur le fichier d'entrée (**stdin**).
Voir commande : pwd

La commande suivante présente toutes les lignes du fichier essai.txt qui contiennent le caractère “**c**” :

```
grep c ~/test1/essai.txt
```

retour de la commande

```
tartempion eric
greg lucien
howard charles
```

Et celle-là pour la lettre “**u**” :

```
grep u ~/test1/essai.txt
```

retour de la commande

```
greg lucien
```

La commande `grep` peut être étendue à tous les fichiers d'un **répertoire**, ou encore à tous les fichiers correspondant à un certain *masque*.

Dans l'exemple qui suit : `grep a te*/*`, la commande `grep` explorera dans le répertoire :

1. /home de l'*user*
2. à l'intérieur des répertoires commençant par “**te**”
3. tous les types de fichiers dont les chaînes de caractère contiennent la lettre “**a**”

```
grep a ~/te*/*
```

retour de la commande

```
test1/essai.txt:tartempion eric
test1/essai.txt:howard charles
test2/adresse.c:25, rue de la source
test2/adresse.c:5bis, avenue linux
test2/adresse.c:3358, street gandhi
```

Les OPTIONS

option -n

L'option `-n` ajoute le numéro de ligne, ainsi :

```
grep -n a ~/te*/*
```

retour de la commande

```
test1/essai.txt:2:tartempion eric
test1/essai.txt:4:howard charles
test2/adresse.c:1:25, rue de la source
test2/adresse.c:2:5bis, avenue linux
```

```
test2/adresse.c:3:3358, street gandhi
```

option -i

L'option `-i` permet d'ignorer la case (ne fait pas la distinction entre les lettres majuscules et minuscules).

option -v

L'option `-v` fait la négation, c'est-à-dire qu'elle affiche toutes les lignes, SAUF celles qui contiennent la chaîne de caractères donnée.

option -c

L'option `-c` ne donne que le nombre de lignes où la chaîne apparaît sans afficher ces lignes.

option -m

L'option `-m N occurrence` : permet d'arrêter de lire un fichier après avoir trouvé N ligne(s) avec occurrence.

Expressions rationnelles

Les expressions rationnelles, ou regexp, peuvent remplacer la chaîne de caractères dans l'argument du programme grep.

Voici des exemples :

Expression	Valeur
abc	Cherche la chaîne abc n'importe où dans la ligne.
^abc	Cherche la chaîne abc en début de ligne.
abc$	Cherche la chaîne abc en fin de ligne.
^abc$	Cherche les ligne ne contenant que abc (commençant et se terminant par abc).
st[a-z][a-z]ic	Cherche n'importe quelle chaîne de caractères commençant par st, suivie de deux lettres minuscules et se terminant par ic (i.e static).
^int * main	Cherche n'importe quelle chaîne de caractères en début de ligne commençant par int, suivie de n'importe quoi et se terminant par argc (i.e int main (int argc, char **argv)).
X[0-2][0-9]	Cherche n'importe quelle chaîne de caractères commençant par X, suivie de 0, 1, ou 2 et se terminant par un chiffre (i.e X11).

Nota :

Programme grep avec expressions rationnelles :

Il faut toutefois prendre un soin particulier lorsque l'on utilise les caractères spéciaux :
`$, *, [, ], ^, |, (, )`
dans l'expression régulière car ces caractères ont une signification particulière pour le shell.
Il vaut mieux mettre l'expression régulière entre apostrophes simples ou doubles **'…'** ou **"…"**
comme ceci par exemple :

```
grep -n "^t" te*/*
```

retour de la commande

```
test1/essai.txt:2:tartempion eric
```

Notez que les champs des lignes produites par `grep` (le nom du fichier, le numéro de ligne et la ligne elle-même) sont délimitées par des deux points "**:**", ce qui leur permet d'être traités à postériori par le programme awk dans des TUBES.

Pour plus d'information sur la commande grep, lancez :

```
man grep
```

Script pratique

La cinémathèque à Jojo

Je suis avec une ligne de texte indiquant les fichiers des films d'une cinémathèque se présentant avec des dossier-titres&fichiers-films alternés.
Je désire créer la liste de l'ensemble.

Je commence par utiliser la commande tree.

Je me retrouve donc avec une liste_films.txt où les titres des dossiers et ceux des films sont indiqués l'un dessous de l'autre, il me faut donc supprimer toutes les lignes des fichiers des films et ne conserver que celles des répertoires, sans extension mp4,

Facile, ils ont tous leur extension propre, mp4.mkv ou VOB.

Trêve de détail, voici la commande qui a conservé la ligne des titres des répertoires et supprimé toutes celles des films :

On se place où se situe le fichier des films titré avec tree, par exemple un fichier qu'on aura nommé liste.txt, et on envoi le bouzin :

```
grep -viE "\.(mkv|mp4|vob)" liste.txt
```

Et il ne nous reste plus qu'à créer le nouveau fichier.txt correctement trié à l'aide des commandes cat et chevron assemblées.

Merci au captnfab qui a tout soufflé le procédé ! 😉

Lien vers le forum

- (Résolu) Cherche commande bash Oui - Non

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

2)

acronyme : Sigle pouvant être prononcé comme un mot. Eh oui !

3)

touch

4)

cat