

# Du bon emploi de grep

**grep** , **egrep** , **fgrep** , **rgrep** , **pgrep** - sont des commandes dans des systèmes d'exploitation de type Unix qui impriment des lignes correspondant à un modèle.

Le **grep** recherche des modèles (**pattern**) sur des fichiers (**input**) et affiche les lignes correspondantes.

En outre, les programmes de variantes **egrep** , **fgrep** et **rgrep** sont identiques à **grep -E** , **grep -F** et **grep -r** respectivement. Ces variantes sont obsolètes, mais sont fournies pour compatibilité avec les versions antérieures.

**Pré requis** Pour les exemples ci dessus, créer un fichier fichier.txt avec ce contenu

```
tendre
vendre
prendre
taper
tasser
masser
casser
caser
attendre
reprendre
surprendre
comprendre
surmonter
surpasser
surévaluer
surdoué
caler
tailler
lister
lisser
```

## 1. grep : recherche de base

### Syntaxe

```
grep LesOptions LeMotif fichier
```

si plusieurs fichiers à rechercher

```
grep LesOptions LeMotif fichier1 fichier2 fichier3
```

## 1.1 Emploi simple

Rechercher le motif 'www' dans /etc/default/grub

```
grep 'GRUB' /etc/default/grub
GRUB_DEFAULT=0
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_LINUX=""
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"
#GRUB_TERMINAL=console
# you can see them in real GRUB with the command `vbeinfo`
#GRUB_GFXMODE=640x480
# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to
Linux
#GRUB_DISABLE_LINUX_UUID=true
#GRUB_DISABLE_RECOVERY="true"
#GRUB_INIT_TUNE="480 440 1"
```

Retrouvera toutes les occurences du motif GRUB

## 1.2 Limiter le motif à un mot précis

Si l'on ne veut que le mot GRUB, il faut pas l'option **-w**

```
grep -w 'GRUB' /etc/default/grub
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
# you can see them in real GRUB with the command `vbeinfo`
# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to
Linux
```

## 1.3 Gestion de la casse

Si l'on ne veut pas tenir compte de la casse, il faut utiliser l'option **-i**

```
$ grep -i 'grub' /etc/default/grub
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# info -f grub -n 'Simple configuration'
GRUB_DEFAULT=0
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_LINUX=""
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
```

```
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xfefefef"
# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console
# you can see them in real GRUB with the command `vbeinfo`
#GRUB_GFXMODE=640x480
# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to
Linux
#GRUB_DISABLE_LINUX_UUID=true
#GRUB_DISABLE_RECOVERY="true"
# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"
```

Cela sortira les motifs : **grub** et **GRUB**

## 1.4 Les ancres (début et fin)

Rechercher les motifs dont les lignes **COMMENCENT** par ; on utilisera le `^ $ egrep "#GRUB" /etc/default/grub #GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xfefefef" #GRUB_TERMINAL=console #GRUB_GFXMODE=640x480 #GRUB_DISABLE_LINUX_UUID=true #GRUB_DISABLE_RECOVERY="true" #GRUB_INIT_TUNE="480 440 1"` Rechercher les motifs dont les lignes **FINISSENT** par ; on utilisera le `$` `bash` `$ grep "console$" /etc/default/grub`

`#GRUB_TERMINAL=console` Rechercher les lignes ne contenant qu'un motif. Mix des 2 situations précédentes: `bash` `$ grep "^leMotif$ /le/fichier/lu` `====`

**1.5 Remplacement de caractères** `==== Utiliser le point pour remplacer un caractère`

`bash` `$ grep 'ca.er' fichier.txt caser caler` Pour complacer, plusieurs caractères, mettre n points avec n=nombre de caractères à remplacer. `bash` `$ grep 'li..er' fichier.txt lister lisser`

Pour remplacer un nombre non défini de caractères faire suivre le point d'un astérix \* `bash` `$ grep 'sur.*er' fichier.txt surmonter surpasser surévaluer` `====`

**1.6 Compter, afficher des numéros de lignes** `====`

Utiliser l'option -c pour compter le nombre d'occurrence `bash` `$ grep -c 'er$' fichier.txt 12` Afficher le numéro des lignes contenant l'occurrence du motif de recherche : option -n `bash` `$ grep -n 'er$' fichier.txt 4:taper 5:tasser 6:masser 7:casser 8:caser 13:surmonter 14:surpasser 15:surévaluer 17:caler 18:tailler 19:lister 20:lisser`

`====` **1.7 Coloriser les occurrences du motif de recherche** `==== Ajouter l'option -color` `bash` `$ grep -color 'er$' fichier.txt` Affichera le motif en rouge sur la sortie écran

Changer la couleur par défaut. `bash` `$ GREP_COLOR='1;32' grep -color 'er$' fichier.txt`

**1.8 Forcer la correspondance inversée** grep Utiliser l'option -v pour imprimer inverse la correspondance; c'est-à-dire qu'il ne correspond qu'aux lignes qui ne contiennent pas le mot donné. Par exemple, imprimez toutes les lignes qui ne contiennent pas le motif lister `bash` `$ egrep -v 'lister' fichier.txt`

On peut bien sûr le faire aussi avec des regex (egrep) `bash` `$ egrep -v '^ca' fichier.txt` Affichera tout ce qui ne commence pas par le motif 'ca' `====`

**2. GREP et LES REGEX (expressions régulières)** `====` **2.1 Le OU (OR) et le ET (AND)** `====`

Pour combiner des motifs : | (équivalent à OU) `bash` `$ egrep "INIT|GFX" /etc/default/grub #GRUB_GFXMODE=640x480 #GRUB_INIT_TUNE="480 440 1"` La sortie contiendra les résultats relatifs aux 2 motifs

Pour obliger 2 motifs .\* (équivalent à AND) `bash` `$ egrep "GRUB.*DEFAULT|DEFAULT.*GRUB" /etc/default/grub GRUB_DEFAULT=0 GRUB_CMDLINE_LINUX_DEFAULT="quiet"`

Retournera les lignes comprenant GRUB et DEFAULT ou DEFAULT Pour utiliser le OR et le AND

avec `grep` et non `egrep`, il faudra préfixer le | avec un antislash : `<code bash>$ grep "INIT\|GFX" /etc/default/grub #GRUB_GFXMODE=640x480 #GRUB_INIT_TUNE="480 440 1" </code>`

==== 2.2 Le crochets [ ] ==== Les expressions entre crochets [] permettent de faire correspondre un groupe de caractères en les mettant entre crochets []. `<code bash>$ egrep 'sur' fichier.txt` surprendre surmonter surpasser surévaluer surdoué `</code>` Retournera tout ce qui contient le motif 'sur'. `<code bash>$ egrep 'sur[dp]' fichier.txt` surprendre surpasser surdoué `</code>` Retournera ce qui commence par le motif 'sur' puis la lettre 'd' ou 'p'. Comme le précédent, mais l'option entre crochet suivi de ^ permet d'éliminer les résultats contenant ensuite les lettres 'd' et 'p'. `<code bash>$ egrep 'sur[^dp]' fichier.txt` surmonter surévaluer `</code>` Au lieu de placer les caractères un par un, vous pouvez spécifier une plage de caractères à l'intérieur des crochets. Une expression de plage est construite en spécifiant les premier et dernier caractères de la plage séparés par un trait d'union. Par exemple, [a-e] est équivalent à [abcde] et [1-3] est équivalent à [123]. `<code bash>$ egrep 'sur[a-f]' fichier.txt` surévaluer surdoué `</code>` Retournera tout ce qui commence par le motif 'sur' puis les lettres a, b, c, d, e ou f. Ainsi pour ne chercher que les lignes commençant par une majuscule : `<code bash>$ grep '^[A-Z]' monFichier.txt</code>` `grep` prend également en charge des classes de caractères prédéfinies qui sont placées entre crochets. En voici la liste : `[[:alnum:]]` | Caractères alphanumériques | `[[:alpha:]]` | Caractères alphabétiques | `[[:blank:]]` | Espaces et tabulations | `[[:digit:]]` | Chiffres | `[[:lower:]]` | Lettres minuscules | `[[:upper:]]` | Lettres majuscules |

==== 2.3 Les quantificateurs ==== Les quantificateurs vous permettent de spécifier le nombre d'occurrences d'éléments qui doivent être présents pour qu'une correspondance se produise. Le tableau suivant montre les quantificateurs pris en charge par `grep` | \* | Faire correspondre l'élément précédent zéro ou plusieurs fois. | ? | Faire correspondre l'élément précédent zéro ou une fois. | + | Faire correspondre l'élément précédent une ou plusieurs fois. | {n} | Faire correspondre exactement à l'élément précédent n fois. | {n,} | Faire correspondre à l'élément précédent au moins n fois. | {, m} | Faire correspondre à l'élément précédent au maximum m fois. | {n, m} | Faire correspondre l'élément précédent de n à m fois. | On peut bien sûr mixer les crochets et les accolades\*\*. Par exemple, recherche de chaînes numériques contenant entre 5 et 7 chiffres:

```
$ egrep '[[[:digit:]]]{3,9}' monFichierAvecDesNombres.txt
```

## 2.4 GREP et la récursivité

Utiliser l'option -R

```
# grep -R 192.168.0. /etc/ssh
/etc/ssh/pssh_host/pssh_hosts:#root@192.168.0.23
/etc/ssh/pssh_host/pssh_hosts:root@192.168.0.11
/etc/ssh/pssh_host/pssh_hosts:root@192.168.0.43
```

La requête de recherche du motif s'effectuera dans le répertoire donné et ses sous-répertoires

Avec des REGEX. Syntaxe identique, mais avec `egrep`

```
# egrep -R '192.168.0.(23|11)' /etc/ssh
/etc/ssh/pssh_host/pssh_hosts:#root@192.168.0.23
```

```
/etc/ssh/pssh_host/pssh_hosts:root@192.168.0.11
```

### 3. Les tubes (pipes)

Utiliser grep avec des pipes pour filtrer les résultats d'une commande

```
$ sudo dmesg | egrep '(s|h)d[a-z]'
```

Triera la sortie de dmesg en ne gardant que les lignes contenant le nom des Diques Durs.

### 4. Mémo

#### Rappel

grep	Expressions régulières de base (BRE)
egrep (grep -E)	Expressions régulières étendues (ERE)
fgrep (grep -F)	Ne recherche que les chaînes
pgrep	Processus de recherche par nom

#### Les principales options de grep

Options de grep	Description
-i	Ignorer la casse
-w	Le motif ne correspond qu'à un mot
-v	Inverser la sélection
-n	Afficher les numéros de lignes des occurrences
-r	Recherche récursive
-R	Identique à -r mais inclut les liens symboliques
-l	Afficher uniquement le nom du fichier contenant l'occurrence
-c	Compter le nombre d'occurrences
-color	Afficher l'occurrence en couleur

Grep plus en détail

```
$ man grep
```

From:  
<https://cbiot.fr/dokuwiki/> - **Cyrille BIOT**

Permanent link:  
<https://cbiot.fr/dokuwiki/grep?rev=1587900539>

Last update: **2020/04/26 13:28**

