

HowTo: Use grep Command In Linux/UNIX – Examples

by Vivek Gite on August 2, 2007 *last updated* December 24, 2014
in [BASH Shell](#), [Linux](#), [UNIX](#)

UNIX

How do I use grep command on Linux, Apple OS X, and Unix-like operating systems? Can you give me a simple examples of the grep command?

The `grep` command is used to search text or searches the given file for lines containing a match to the given strings or words. By default, `grep` displays the matching lines. Use `grep` to search for lines of text that match one or many regular expressions, and outputs only the matching lines. `grep` is considered as one of the most useful commands on Linux and Unix-like operating systems.

Did you know?

The name, "grep", derives from the command used to perform a similar operation, using the Unix/Linux text editor `ed`:

```
g/re/p
```

The grep command syntax

The syntax is as follows:

```
grep 'word' filename
grep 'word' file1 file2 file3
grep 'string1 string2' filename
cat otherfile | grep 'something'
command | grep 'something'
command option1 | grep 'data'
grep --color 'data' fileName
```

How do I use grep command to search a file?

Search `/etc/passwd` file for `boo` user, enter:

```
$ grep boo /etc/passwd
```

Sample outputs:

```
foo:x:1000:1000:foo,,,:/home/foo:/bin/ksh
```

You can force `grep` to ignore word case i.e match `boo`, `Boo`, `BOO` and all other combination with the `-i` option:

```
$ grep -i "boo" /etc/passwd
```

Use grep recursively

You can search recursively i.e. read all files under each directory for a string "192.168.1.5"

```
$ grep -r "192.168.1.5" /etc/
```

OR

```
$ grep -R "192.168.1.5" /etc/
```

Sample outputs:

```
/etc/ppp/options:# ms-wins 192.168.1.50
/etc/ppp/options:# ms-wins 192.168.1.51
/etc/NetworkManager/system-connections/Wired connection
1:addresses1=192.168.1.5;24;192.168.1.2;
```

You will see result for 192.168.1.5 on a separate line preceded by the name of the file (such as /etc/ppp/options) in which it was found. The inclusion of the file names in the output data can be suppressed by using the `-h`

option as follows:

```
$ grep -h -R "192.168.1.5" /etc/
```

OR

```
$ grep -hR "192.168.1.5" /etc/
```

Sample outputs:

```
# ms-wins 192.168.1.50
# ms-wins 192.168.1.51
addresses1=192.168.1.5;24;192.168.1.2;
```

Use grep to search words only

When you search for boo, grep will match fooboo, boo123, barfoo35 and more. You can force the grep command to select only those lines containing matches that form whole words i.e. match only boo word:

```
$ grep -w "boo" file
```

Use grep to search 2 different words

Use the egrep command as follows:

```
$ egrep -w 'word1|word2' /path/to/file
```

Count line when words has been matched

The grep can report the number of times that the pattern has been matched for each file using `-c` (count) option:

```
$ grep -c 'word' /path/to/file
```

Pass the `-n` option to precede each line of output with the number of the line in the text file from which it was obtained:

```
$ grep -n 'root' /etc/passwd
```

Sample outputs:

```
1:root:x:0:0:root:/root:/bin/bash
1042:rootdoor:x:0:0:rootdoor:/home/rootdoor:/bin/csh
3319:initrootapp:x:0:0:initrootapp:/home/initroot:/bin/ksh
```

Grep invert match

You can use `-v` option to print inverts the match; that is, it matches only those lines that do not contain the given word. For example print all line that do not contain the word bar:

```
$ grep -v bar /path/to/file
```

UNIX / Linux pipes and grep command

grep command often used with [shell pipes](#). In this example, show the name of the hard disk devices:

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

Display cpu model name:

```
# cat /proc/cpuinfo | grep -i 'Model'
```

However, above command can be also used as follows without shell pipe:

```
# grep -i 'Model' /proc/cpuinfo
```

Sample outputs:

```
model          : 30
model name     : Intel(R) Core(TM) i7 CPU           Q 820  @ 1.73GHz
model          : 30
model name     : Intel(R) Core(TM) i7 CPU           Q 820  @ 1.73GHz
```

How do I list just the names of matching files?

Use the -l option to list file name whose contents mention main():

```
$ grep -l 'main' *.c
```

Finally, you can force grep to display output in colors, enter:

```
$ grep --color vivek /etc/passwd
```

Sample outputs:

```
vivek@wks01:~$ grep --color vivek /etc/passwd
vivek:x:1000:1000:Vivek Gite,,,:/home/vivek:/bin/bash
vivek@wks01:~$ █
```

Grep command in action