Using grep to find string in files

Article Number: 683 | Rating: Unrated | Last Updated: Fri, May 15, 2020 12:18 PM

Using grep to find string in files

How many times I had to search for something, let’s say my mac address, to change it in all files under /etc ? I couldn’t remebmer, different distros – different files, anyway, that’s an example, there are many out there, like searching log files, strings like IPs or titles and the worst part, recursively. I will show few examples of Using grep to find string in files and recursively too.

You need to use the grep command. The grep command searches the given input files for lines containing a match or a text string.

grep command syntax

The syntax is:

```
grep "text string to search” directory-path
```

OR

```
grep [option] "text string to search” directory-path
```

OR

```
grep -r "text string to search” directory-path
```

OR

```
grep -r -H "text string to search” directory-path
```

OR

```
egrep -R "word-1|word-2” directory-path
```

OR

```
egrep -w -R "word-1|word-2” directory-path
```

Examples

In this example, search for a string called ‘redeem reward’ in all text (*.txt) files located in /home/tom/ directory, use:

```
$ grep "redeem reward" /home/tom/*.txt
```

OR

```
$ grep "redeem reward" ~/*.txt
```

Task: Search all subdirectories recursively

You can search for a text string all files under each directory, recursively with -r option:

```
$ grep -r "redeem reward" /home/tom/
```

OR

```
$ grep -R "redeem reward" /home/tom/
```

Task: Only display filenames

By default, the grep command prints the matching lines. You can pass -H option to print the filename for each match:

```
$ grep -H -r "redeem reward" /home/tom
```

Sample outputs:

To just display the filename use the cut command as follows:

```
$ grep -H -R vivek /etc/* | cut -d: -f1
```

Sample outputs:

Task: Suppress file names

The grep command shows output on a separate line, and it is preceded by the name of the file in which it was found in the case of multiple files. You can pass the -h option to suppress inclusion of the file names in the output:

```
$ grep -h -R 'main()' ~/projects/*.c
```

Task: Display only words

You can select only those lines containing matches that form whole words using the -w option. In this example, search for word ‘getMyData()’ only in ~/projects/ dirctory:

```
$ grep -w -R 'getMyData()' ~/projects/
```

Task: Search for two or more words

Use the egrep command as follows:

```
$ egrep -w -R 'word1|word2' ~/projects/
```

Task: Hide warning spam

grep command generate error message as follows due to permission and other issues:

```
No such file or directory
No such device or address
Permission denied
```

To hide all errors or warning message spam generated by the grep command, append 2>/dev/null to grep command. This will send and hide unwanted output to /dev/null device:

$ grep -w -R 'getMyData()' ~/projects/ 2>/dev/null

Task: Display matched text in color

Pass the --color option to the grep command display matched text/words in color on the terminal:

[redacted]

[redacted]

Task: Ignore case (my favourite since the last grep update)

Our final example ignore case distinctions in both the search PATTERN and the input files:

```
grep -s -R 'word' /path/to/dir
grep -s -r 'income tax' ~/accounting/
```

or just

```
grep -sR ‘word’ /path/
```

Posted - Fri, May 15, 2020 12:18 PM. This article has been viewed 2116 times.

Online URL: http://kb.ictbanking.net/article.php?id=683