

# How to use the grep command on Debian 10

Grep stands for Global Regular Expression Print. This is a useful command and is widely used by Linux system engineers, while searching for a string or patterns with regular files and in the system.

**Grep** stands for Global Regular Expression Print. This is a useful command and is widely used by Linux system engineers, while searching for a string or patterns with regular files and in the system.

In this article, **TipsMake.com** will show you how to use **grep** command (many examples are included).

**Note** : The article has tested all commands and examples on machines running Debian 10.

## Do you know how to use grep?

1. Prerequisites
2. Install grep on Debian 10
3. Use the grep command
  1. Search for a specific file or directory in the system
  2. Search for a complete word with grep
  3. Search for a specific piece of text in a file
  4. Do a search in multiple files
  5. Search for two different words with a single grep command
  6. Number the lines that fit the text
  7. Reverse search
  8. Count matches
  9. Displays file names that match specific text
  10. Show only matching text
  11. Display lines starting with specific word (s)
  12. Display lines that end with specific word (s)

## Prerequisites

You need to have a computer running Debian 10 with root access.

## Install grep on Debian 10

By default, **grep** is installed on most systems including Debian 10. If it is not already installed, open the terminal and issue the following command as root.

```
apt-get install grep
```

When you are asked to confirm, press Y and then type from the keyboard. Wait for the installation process to finish.

Check the version of grep by running the following command in the terminal.

```
grep --version
```

The command will return the version along with other details as shown below.

A terminal window showing the output of the 'grep --version' command. The prompt is 'root@debian:/home/karim#'. The output is: 'grep (GNU grep) 3.3', 'Copyright (C) 2018 Free Software Foundation, Inc.', 'License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.', 'This is free software: you are free to change and redistribute it.', 'There is NO WARRANTY, to the extent permitted by law.', 'Written by Mike Haertel and others; see <https://git.sv.gnu.org/cgiit/grep.git/tree/AUTHORS>.', followed by several more prompts 'root@debian:/home/karim#'.

The command will return the version along with other details

## Use the grep command

### Search for a specific file or directory in the system

When you want to search or locate a specific file in your system, the command's syntax is as follows.

```
ls -l ???ng d?n c?a th? m?c mu?n tîm ki?m> | grep -i 'tên file ho?c th? m?c'
```

Need to put the word you want to find in quotation marks if it contains spaces. Assuming you are looking for **'network daemon'** , the command will look like the following.

```
ls -l /etc/network/ | grep -i 'interfaces daemon'
```

### Search for a complete word with grep

You may notice, **grep** has returned all sorts of results with the word **'network'** , for example networks, networked, networking or abcnetworking, etc. If you want to limit your search to include specific words If so, you must use the **-w** option as follows.

```
ls -l /etc/ | grep -i -w network
```

### Search for a specific piece of text in a file

Suppose you have a large file and you want to search for a specific piece of text. The syntax of the command will be as follows.

```
grep -i 'v?n b?n b?n mu?n tìm ki?m' 'tên file và ???ng d?n'
```

## Do a search in multiple files

If you want to search for a document from a large number of files and subfolders in a large directory, you can do this using the **-r** option .

```
grep -i -r "fox"
```

Here is a sample output showing the word **'fox'** included in both **test.txt** and **tree.txt** :

```
karim@debian:~$  
karim@debian:~$  
karim@debian:~$ grep -i -r fox  
tree.txt:dasdadasdas dasd asd as d asd as das d fox  
test.txt:Lazy fox jump over the dog. It was hidden away somewhere in a forest. No one was watching a  
nd therefore this is the story. All I can say we have won the game. That's all I can say.  
karim@debian:~$  
karim@debian:~$  
karim@debian:~$
```

The word 'fox' is found in both test.txt and tree.txt

You can also provide a directory path and it will search all the files in that directory and its subfolders.

Suppose you want to find **'interfaces'** in **/etc/** and its subfolders. The command should be as follows.

```
grep -i -r interfaces /etc/
```

Here is the sample output:

```
/etc/default/networking#CONFIGURE_INTERFACES=yes  
/etc/default/networking# Don't configure these interfaces, shell wildcards supported/  
/etc/default/networking#EXCLUDE_INTERFACES=  
/etc/default/networking# If none given, wait for all auto interfaces, or if there are none,  
grep: /etc/shadow: Permission denied  
grep: /etc/passwd: Permission denied  
/etc/osprow.d/abstractions/base: # Sometimes used to determine kernel/user interfaces to  
/etc/sysctl.conf# Turn on Source Address Verification in all interfaces to  
grep: /etc/ssh/ssh_host_ecdsa_key: Permission denied  
grep: /etc/ssh/ssh_host_ed25519_keys: Permission denied  
grep: /etc/ssh/ssh_host_rsa_key: Permission denied  
grep: /etc/security/openssl: Permission denied  
grep: /etc/shadow: Permission denied  
/etc/init.d/networking# Short-Description: Raise network interfaces.  
/etc/init.d/networking# Description: Prepare /run/network directory, ifstate file and raise n  
etwork interfaces, or tag them down.  
/etc/init.d/networking#CONFIGURE_INTERFACES=yes  
/etc/init.d/networking#EXCLUDE_INTERFACES=  
/etc/init.d/networking: set -- EXCLUDE_INTERFACES  
/etc/init.d/networking: log_warning_msg "not deconfiguring network interfaces: ISCSI root is mounted  
"  
/etc/init.d/networking: log_warning_msg "not deconfiguring network interfaces: network devices s  
till mounted."  
/etc/init.d/networking: log_warning_msg "not deconfiguring network interfaces: network file syst  
em still mounted."  
/etc/init.d/networking: log_warning_msg "not deconfiguring network interfaces: network swap still  
1 mounted."  
/etc/init.d/networking: if [ "$CONFIGURE_INTERFACES" = no ]  
/etc/init.d/networking: log_action_msg "not configuring network interfaces, see /etc/default/net  
working"  
/etc/init.d/networking: log_action_begin_msg "Configuring network interfaces"  
/etc/init.d/networking: log_action_begin_msg "deconfiguring network interfaces"  
/etc/init.d/networking: log_action_begin_msg "Reloading network interfaces configuration"  
/etc/init.d/networking: log_warning_msg "Running $0 $1 is deprecated because it may not re-enable so  
me interfaces"  
/etc/init.d/networking: log_action_begin_msg "Reconfiguring network interfaces"  
karim@debian:~$
```

Do a search in multiple files

## Search for two different words with a single grep command

You can search for two different words with an **egrep** command (a variant of **grep** ) as follows. Let's say you want to search for **fox** and **lazy words** in multiple files with the **-r** option . You must run the following command on the terminal.

```
egrep -w -r "fox|lazy"
```

Here is the sample output:

```
karim@debian:~$  
karim@debian:~$  
karim@debian:~$ egrep -w -n 'fox|lazy'  
calm.txt:lazy  
test.txt:Lazy fox jump over the dog. It was hidden away somewhere in a forest. No one was watching a  
nd therefore this is the story. All I can say we have won the game. That's all I can say.  
karim@debian:~$  
karim@debian:~$  
karim@debian:~$  
karim@debian:~$
```

Search for two different words with a single grep command

## Number the lines that fit the text

Another useful option is **-n** numbering of lines of text. The following is an example illustrating how to use the **-n** option .

```
grep -i -n "fox" test.txt
```

The following is a sample output for the numbering of lines matching the word **'fox'** .

```
karim@debian:~$ grep -i -n fox test.txt  
1:Lazy fox jump over the dog. It was hidden away somewhere in a forest.  
4:That's all I can say fox.  
8:so this is the fox  
karim@debian:~$  
karim@debian:~$  
karim@debian:~$  
karim@debian:~$
```

Number the lines that fit the text

## Reverse search

This is in contrast to what we did above. If you want to return a text that does not include the word you specify, you can use the **-v** option .

The following is an example that illustrates the use of the **-v** option .

```
grep -v -i "fox" test.txt
```

Here is the sample output:

```
karim@debian:~$  
karim@debian:~$  
karim@debian:~$  
karim@debian:~$  
karim@debian:~$  
karim@debian:~$ grep -v -i fox test.txt  
No one was watching and therefore this is the story.  
All I can say we have won the game.  
we all hav to go one day.  
This is the reality.  
No one will live here forever.  
karim@debian:~$
```

## Reverse search

All of the above options ( **-n** , etc.) can also be applied with the **-v** option .

## Count matches

If you want to count the number of results that match a specific text, you can use the **-c** option .

Count the word '**fox**' in **test.txt** inside the current directory. Run the following command in the terminal:

```
grep -i -c fox test.txt
```

Below is a sample output after executing the above command, showing that the word '**fox**' has appeared 3 times in the **test.txt** file .



```
karim@debian:~$  
karim@debian:~$  
karim@debian:~$  
karim@debian:~$ grep -i -c fox test.txt  
3  
karim@debian:~$  
karim@debian:~$
```

## Count matches

## Displays file names that match specific text

If you want to find files containing your specific words, you can use the **-l** option with **-r** as follows.

Assuming all the files are located in the current directory and the specific word you are looking for is '**fox**'.

```
grep -i -r -l fox
```

Here is a sample output showing the word '**fox**' is present in **test.txt** , as well as in the subdirectory and **asif.txt** file:

Displays file names that match specific text



```
karim@debian:~$  
karim@debian:~$  
karim@debian:~$  
karim@debian:~$ grep -i -r -l fox  
test/asif.txt  
test.txt  
karim@debian:~$  
karim@debian:~$  
karim@debian:~$
```

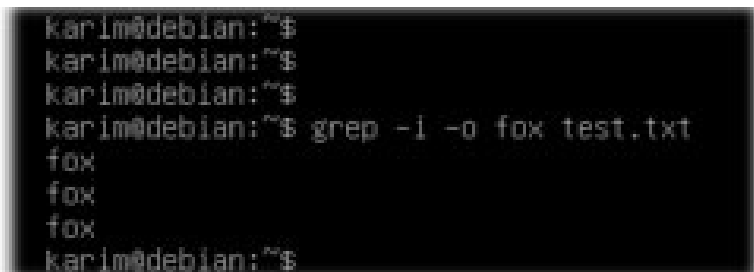
Displays file names that match specific text

## Show only matching text

By default, **grep** displays entire lines that match the text or words you want. If you want **grep** to display words that match, you can use the **-o** option as follows.

```
grep -i -o fox test.txt
```

Here is a sample output.



```
karim@debian:~$  
karim@debian:~$  
karim@debian:~$  
karim@debian:~$ grep -i -o fox test.txt  
fox  
fox  
fox  
karim@debian:~$
```

Show only matching text

## Display lines starting with specific word (s)

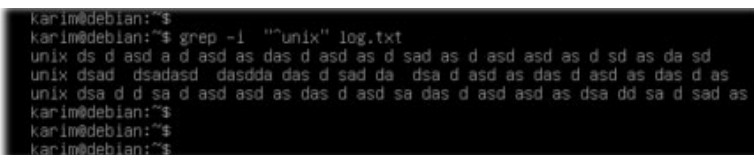
If you want to retrieve all lines starting with a specific word, you can use the **^** operator as follows.

Suppose you want to return all lines beginning with **'unix'** and the file is **log.txt** located in the current directory.

Run the following command in the terminal:

```
grep -i "^unix" log.txt
```

Here is the sample output:



```
karim@debian:~$  
karim@debian:~$ grep -i "^unix" log.txt  
unix ds d asd a d asd as das d asd as d sad as d asd asd as d sd as da sd  
unix dsad dsadasd dsasdda das d sad da dsa d asd as das d asd as das d as  
unix dsa d d sa d asd asd as das d asd sa das d asd asd as dsa dd sa d sad as  
karim@debian:~$  
karim@debian:~$  
karim@debian:~$
```

Display lines starting with specific word (s)

## Display lines that end with specific word (s)

If you want to return all lines from a file ending with a specific word (s), you can use the **\$** operator as follows.

Assuming the word is **'linux'** and the file you want to search for is **rev.txt** located in the current directory.

Run the following command:

```
grep -i "linux$" rev.txt
```

Here is the sample output:

```
karim@debian:~$ grep -i "linux$" rev.txt
yydsad dsdasd asdhda dhdas sadhads dasda linux
dsad dsa dasd as das dsad as das d as das das d linux
dasd euqsadas csadsad wqeadssad dasda das d das linux
karim@debian:~$
karim@debian:~$
```

Display lines that end with specific word (s)

You finished reading the article "**How to use the grep command on Debian 10**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.