grep command in Unix/Linux

The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression (grep stands for global search for regular expression and print out).

Syntax:

```
grep [options] pattern [files]
```

```
Options Description
-c : This prints only a count of the lines that match a pattern
-h : Display the matched lines, but do not display the filenames.
-i : Ignores, case for matching
-l : Displays list of a filenames only.
-n : Display the matched lines and their line numbers.
-v : This prints out all the lines that do not matches the pattern
-e exp : Specifies expression with this option. Can use multiple times.
-f file : Takes patterns from file, one per line.
-E : Treats pattern as an extended regular expression (ERE)
-w : Match whole word
-o : Print only the matched parts of a matching line,
 with each such part on a separate output line.

-A n : Prints searched line and nlines after the result.
-B n : Prints searched line and n line before the result.
-C n : Prints searched line and n lines after before the result.
```

Consider the below file as an input.

```
$cat > geekfile.txt
```

```
unix is great os. unix was developed in Bell labs.
learn operating system.
Unix linux which one you choose.
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
```

1. Case insensitive search : The -i option enables to search for a string case insensitively in the given file. It matches the words like “UNIX”, “Unix”, “unix”.

```
$grep -i "UNix" geekfile.txt
```

Output:

```
unix is great os. unix was developed in Bell labs.
Unix linux which one you choose.
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
```

2. Displaying the count of number of matches : We can find the number of lines that matches the given string/pattern

Output:

```
2
```

3. Display the file names that matches the pattern : We can just display the files that contains the given string/pattern.

```
$grep -l "unix" *

or

$grep -l "unix" f1.txt f2.txt f3.xt f4.txt
```

Output:

```
geekfile.txt
```

4. Checking for the whole words in a file : By default, grep matches the given string/pattern even if it is found as a substring in a file. The -w option to grep makes it match only the whole words.

```
$ grep -w "unix" geekfile.txt
```

Output:

```
unix is great os. unix was developed in Bell labs.
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
```

5. Displaying only the matched pattern : By default, grep displays the entire line which has the matched string. We can make the grep to display only the matched string by using the -o option.

```
$ grep -o "unix" geekfile.txt
```

Output:

```
unix
unix
unix
unix
unix
unix
```

6. Show line number while displaying the output using grep -n : To show the line number of file with the line matched.

```
$ grep -n "unix" geekfile.txt
```

Output:

```
1:unix is great os. unix is free os.
4:uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a
powerful.
```

7. Inverting the pattern match : You can display the lines that are not matched with the specified search string pattern using the -v option.

```
$ grep -v "unix" geekfile.txt
```

Output:

```
learn operating system.
Unix linux which one you choose.
```

8. Matching the lines that start with a string : The ^ regular expression pattern specifies the start of a line. This can be used in grep to match the lines which start with the given string or pattern.

```
$ grep "^unix" geekfile.txt
```

Output:

```
unix is great os. unix is free os.
```

9. Matching the lines that end with a string : The $ regular expression pattern specifies the end of a line. This can be used in grep to match the lines which end with the given string or pattern.

```
$ grep "os$" geekfile.txt
```

10.Specifies expression with -e option. Can use multiple times :

```
$grep –e "Agarwal" –e "Aggarwal" –e "Agrawal" geekfile.txt
```

11. -f file option Takes patterns from file, one per line.

```
$cat pattern.txt

Agarwal
Aggarwal
Agrawal



$grep –f pattern.txt  geekfile.txt
```

12. Print n specific lines from a file: -A prints the searched line and n lines after the result, -B prints the searched line and n lines before the result, and -C prints the searched line and n lines after and before the result.

Syntax:

```
$grep -A[NumberOfLines(n)] [search] [file]

$grep -B[NumberOfLines(n)] [search] [file]

$grep -C[NumberOfLines(n)] [search] [file]
```

Example:

```
$grep -A1 learn geekfile.txt
```

Output:

```
learn operating system.
Unix linux which one you choose.
--
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

(Prints the searched line along with the next n lines (here n = 1 (A1).)
(Will print each and every occurrence of the found line, separating each
output by --)
(Output pattern remains the same for -B and -C respectively)
Unix linux which one you choose.
--
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
Unix linux which one you choose.
--
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
```

13. Search recursively for a pattern in the directory: -R prints the searched pattern in the given directory recursively in all the files.

Syntax

```
$grep -R [Search] [directory]
```

Example :

```
$grep -iR geeks /home/geeks
```

Output:

```
./geeks2.txt:Well Hello Geeks
./geeks1.txt:I am a big time geek
----------------------------------
-i to search for a string case insensitively
-R to recursively check all the files in the directory.
```

This article is contributed by **Akshay Rajput**. If you like GeeksforGeeks and would like to contribute, you can also write an article using write.geeksforgeeks.org or mail your article to review-team@geeksforgeeks.org. See your article appearing on the GeeksforGeeks' main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Last Updated : 15 Nov, 2022

167

Previous

Next

Article Contributed By :

GeeksforGeeks

Vote for difficulty

Current difficulty : Easy

Easy Normal Medium Hard Expert

Article Tags : linux-command, Linux-misc-commands, Linux-Unix

Improve Article

Report Issue

feedback@geeksforgeeks.org

Company

About Us

Careers

In Media

Contact Us

Terms and Conditions

Privacy Policy

Copyright Policy

Third-Party Copyright Notices

Advertise with us

Languages

Python

Java

C++

GoLang

SQL

R Language

Android Tutorial

Data Structures

Array

String

Linked List

Stack

Queue

Tree

Graph

Algorithms

Sorting

Searching

Greedy

Dynamic Programming

Pattern Searching

Recursion

Backtracking

Web Development

HTML

CSS

JavaScript

Bootstrap

ReactJS

AngularJS

NodeJS

Write & Earn

Write an Article

Improve an Article

Pick Topics to Write

Write Interview Experience

Internships

Video Internship

Computer Science

GATE CS Notes

Data Science & ML

Data Science With Python

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Maths For Machine Learning

Pandas Tutorial

NumPy Tutorial

NLP Tutorial

Interview Corner

Company Preparation

Preparation for SDE

Company Interview Corner

Experienced Interview

Internship Interview

Competitive Programming

Aptitude

Python

Python Tutorial

Python Programming Examples

Django Tutorial

Python Projects

Python Tkinter

OpenCV Python Tutorial

GfG School

CBSE Notes for Class 8

CBSE Notes for Class 9

CBSE Notes for Class 10

CBSE Notes for Class 11

CBSE Notes for Class 12

English Grammar

UPSC/SSC/BANKING

SSC CGL Syllabus

SBI PO Syllabus

IBPS PO Syllabus

UPSC Ethics Notes

UPSC Economics Notes

UPSC History Notes