**Grep Commands**

The **grep** command in Unix/Linux is a powerful tool used for searching and manipulating text patterns within files. Its name is derived from the ed (editor) command g/re/p (globally search for a regular expression and print matching lines), which reflects its core functionality. grep is widely used by programmers, system administrators, and users alike for its efficiency and versatility in handling text data.

Syntax:

**grep [options] pattern [files]**  
  
[**options**]: These are command-line flags that modify the behaviour of grep.

[**pattern**]: This is the regular expression you want to search for.

[**file**]: This is the name of the file(s) that we want to search within. We can specify multiple files for simultaneous searching

| **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. |

**Practical Examples:**

**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

**2. Displaying the Count of Number of Matches Using grep**

We can find the number of lines that matches the given string/pattern

* grep -c "unix" geekfile.txt

**3. Display the File Names that Matches the Pattern Using grep**

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

**4. Checking for the Whole Words in a File Using grep**

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

**Displaying only the matched pattern Using grep**

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

**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

**7. Inverting the Pattern Match Using grep**

We can display the lines that are not matched with the specified search string pattern using the -v option.

* grep -v "unix" geekfile.txt

**8. Matching the Lines that Start with a String Using grep**

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

**9. Matching the Lines that End with a String Using grep**

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 Using grep**

-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

**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