**Grep Command in Unix and Linux Examples**

Grep is the frequently used command in Unix (or Linux). Most of us use grep just for finding the words in a file. The power of grep comes with using its options and regular expressions. You can analyze large sets of log files with the help of grep command.  
  
Grep stands for Global search for Regular Expressions and Print.  
  
**The basic syntax of grep command is**  
  
grep [options] pattern [list of files]  
  
Let see some practical examples on grep command.  
  
**1.** Running the last executed grep command  
  
This saves a lot of time if you are executing the same command again and again.

!grep

This displays the last executed grep command and also prints the result set of the command on the terminal.  
  
**2.** Search for a string in a file  
  
This is the basic usage of grep command. It searches for the given string in the specified file.

grep "Error" logfile.txt

This searches for the string "Error" in the log file and prints all the lines that has the word "Error".  
  
**3.** Searching for a string in multiple files.

grep "string" file1 file2

grep "string" file\_pattern

This is also the basic usage of the grep command. You can manually specify the list of files you want to search or you can specify a file pattern (use regular expressions) to search for.  
  
**4.** Case insensitive search  
  
The -i option enables to search for a string case insensitively in the give file. It matches the words like "UNIX", "Unix", "unix".

grep -i "UNix" file.txt

**5.** Specifying the search string as a regular expression pattern.

grep "^[0-9].\*" file.txt

This will search for the lines which starts with a number. Regular expressions is huge topic and I am not covering it here. This example is just for providing the usage of regular expressions.  
  
**6.** Checking for the whole words in a file.  
  
By default, grep matches the given string/pattern even if it found as a substring in a file. The -w option to grep makes it match only the whole words.

grep -w "world" file.txt

**7.** Displaying the lines before the match.  
  
Some times, if you are searching for an error in a log file; it is always good to know the lines around the error lines to know the cause of the error.

grep -B 2 "Error" file.txt

This will prints the matched lines along with the two lines before the matched lines.  
  
**8.** Displaying the lines after the match.

grep -A 3 "Error" file.txt

This will display the matched lines along with the three lines after the matched lines.  
  
**9.** Displaying the lines around the match

grep -C 5 "Error" file.txt

This will display the matched lines and also five lines before and after the matched lines.  
  
**10.** Searching for a sting in all files recursively  
  
You can search for a string in all the files under the current directory and sub-directories with the help -r option.

grep -r "string" \*

**11.** Inverting the pattern match  
  
You can display the lines that are not matched with the specified search sting pattern using the -v option.

grep -v "string" file.txt

**12.** Displaying the non-empty lines  
  
You can remove the blank lines using the grep command.

grep -v "^$" file.txt

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

grep -c "sting" file.txt

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

grep -l "string" \*

**15.** Display the file names that do not contain the pattern.  
  
We can display the files which do not contain the matched string/pattern.

grep -L "string" \*

**16.** 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 "string" file.txt

**17.** Displaying the line numbers.  
  
We can make the grep command to display the position of the line which contains the matched string in a file using the -n option

grep -n "string" file.txt

**18.** Displaying the position of the matched string in the line  
  
The -b option allows the grep command to display the character position of the matched string in a file.

grep -o -b "string" file.txt

**19.** 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 "^start" file.txt

**20.** 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 "end$" file.txt