Tutorial Link https://course.testpad.chitkara.edu.in/tutorials/Linux Filters and Shell/6305a6085611a5348833e4ee

TUTORIAL

Linux Filters and Shell

Topics

1.1 grep

1.2 sed

1.3 working of Shell's Wild cards, Escaping and Quoting.

1.4 How to change Shell in Linux

grep

grep Filter in Linux

Introduction

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 n lines 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.

1. Case insensitive search :

```
grep – i “root” /etc/passwd
```

2. Displaying the count of number of matches

```
grep  –c “root” /etc/passwd
```

3. Display the file names that matches the pattern

```
grep  –l “root” *
```

4. Checking for the whole words in a file

```
grep -w “root” /etc/passwd
```

5. Displaying only the matched pattern

```
grep -o “root” /etc/passwd
```

6. Show line number while displaying the output

```
grep –n “root” /etc/passwd
```

7. Inverting the pattern match

```
grep –v “root” /etc/passwd
```

8. Matching the lines that start with a string

```
grep “^root” /etc/passwd
```

9. Matching the lines that end with a string

```
grep “login$” /etc/passwd
```

10. Search recursively for a pattern in the directory

```
grep –iR root /etc/passwd
```

sed

SED command in UNIX stands for stream editor and it can perform lots of functions on file like searching, find and replace, insertion or deletion. Though most common use of SED command in UNIX is for substitution or for find and replace. By using SED you can edit files even without opening them, which is much quicker way to find and replace something in file, than first opening that file in VI Editor and then changing it.

Syntax:

```
sed OPTIONS... [SCRIPT] [INPUTFILE...]
```

Examples:

1 Replaces the word “Unix” with “Linux” in the file :

```
sed 's/Unix/Linux/’ file.txt
```

2 Replacing the nth occurrence of a pattern in a line

```
sed 's/unix/linux/2’ file.txt
```

3 Replacing all the occurrence of the pattern in a line

```
sed 's/unix/linux/g’ file.txt
```

4 Replacing from nth occurrence to all occurrences in a line

```
sed 's/unix/linux/3g’ file.txt
```

5 Replacing string on a specific line number

```
sed '3 s/unix/linux/' file.txt
```

6 Duplicating the replaced line with /p flag

```
sed 's/unix/linux/p' file.txt
```

7 Printing only the replaced lines

```
sed -n 's/unix/linux/p' file.txt
```

8 Replacing string on a range of lines

```
sed '1,3 s/unix/linux/' file.txt
```

9 Deleting lines from a particular file

```
sed '5d' filename.txt (Delete 5th Line)
```

10 Delete last line

```
sed '$d' filename.txt (Delete last line)
```

11 Delete line 5 to last line

```
sed  '5,$d' filename.txt
```

12 Delete 3-6 lines from a file

```
sed '3,6d' filename.txt
```

13 Delete line 12 to last line from a file

```
sed '12,$d' filename.txt
```

working of Shell's Wild cards, Escaping and Quoting.

a)Wild cards:

There are six wild card characters are **:-*,?,[abc],[!abc],[a-z],[!a-z]**

- By the use of this wild card character all the file is select.

```
root@kali:~/dir1# ls
file1  file12  file1qw  file2  fileab  filexy
root@kali:~/dir1# rm file*
root@kali:~/dir1# ls
root@kali:~/dir1#
```

· By the use of this wild card character only one character is select after the file name. If we use ?? then two character are select of alphanumeric.

```
root@kali:~/dir2# ls
file1  file22  file234  file2ad  filexyz
root@kali:~/dir2# rm file?
root@kali:~/dir2# ls
file22  file234  file2ad  filexyz
root@kali:~/dir2# rm file??
root@kali:~/dir2# ls
file234  file2ad  filexyz
root@kali:~/dir2# rm file?ad
root@kali:~/dir2# ls
file234  filexyz
root@kali:~/dir2# rm file???
root@kali:~/dir2# ls
root@kali:~/dir2#
```

· By the use of this wild card only the element which we enter in the character class are select after the filename.

```
root@kali:~/dir3# ls
file1  file2  file3  file4  file6
root@kali:~/dir3# rm file[234]
root@kali:~/dir3# ls
file1  file6
root@kali:~/dir3#
```

· By the use of this wild card only the element which we enter in the character class are not select after the filename.

```
root@kali:~/dir3# ls
file1  file2  file3  file4  file5  file6  file7
root@kali:~/dir3# rm file[!345]
root@kali:~/dir3# ls
file3  file4  file5
root@kali:~/dir3#
```

· By the use of this wild card only the element range which we enter in the character class are select after the filename.

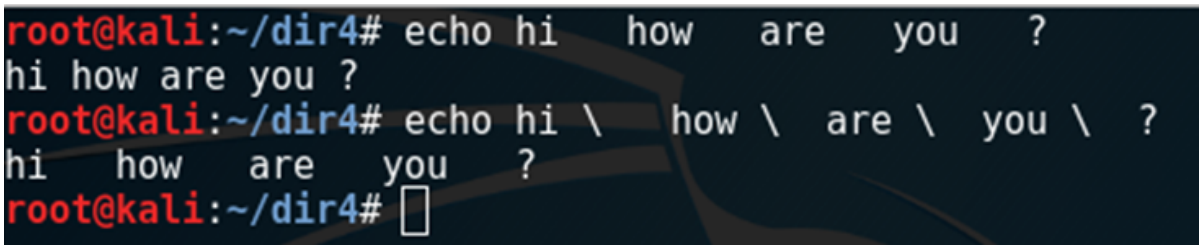
· By the use of this wild card only the element range which we enter in the character class are not select after the filename.

```
root@kali:~/dir4# ls
file1  file2  file3  file4  file5  file6  file7  file8
root@kali:~/dir4# rm file[1-3]
root@kali:~/dir4# ls
file4  file5  file6  file7  file8
root@kali:~/dir4# rm file[!5-7]
root@kali:~/dir4# ls
file5  file6  file7
root@kali:~/dir4#
```

b) Escaping: In this if we want to remove the meaning of the special character then we use escaping.We use **" ",' ',** to remove the special meaning of special character.

```
root@kali:~/dir4# ls
file1  file2  file3  file4  file5  file6  file7  file8
root@kali:~/dir4# rm "file*"
rm: cannot remove 'file*': No such file or directory
root@kali:~/dir4# rm 'file*'
rm: cannot remove 'file*': No such file or directory
root@kali:~/dir4# rm file\*
rm: cannot remove 'file*': No such file or directory
root@kali:~/dir4#
```

c) Quoting : We use this if we want the required space between the word which we enter in the line .For this we use this \ .



How to change Shell in Linux

Change Shell in Linux Servers:

1. First check all available shells in your Linux Servers

```
#cat /etc/shells
```

2. We can change shell through usermod command as under

```
#usermod -s /bin/tcsh username
```

3. We can change shell through chsh command as under

```
 #chsh --shell /bin/tcsh username
```

4. We can also change shell through /etc/passwd file as under

```
#vi /etc/passwd
```

change the shell /bin/bash to /bin/tcsh.With this file we can also directly rename any user or group also.