**UNIX COMMANDS**

**Filters in Linux**

Cat,head,tail,sort,uniq,wc,grep,tac,sed,nl.

**Basic Commands**

**Command**:- who

**Syntax**:- $who

**Explanation**:- It displays details of all users who are currently logged on to the system.

**options**: -H option provides headers for the columns

-u option provides more details like idle time

**Command**: -who am i

**Syntax**: - $ who am i

**Explanation**: -It displays the login name, terminal number and date and time.

**Command**:- man

**Syntax**:- $man <cmdname>

**Explanation**:- man command is short for manual and it gives the details of every command available in UNIX.

**Command**:- passwd

**Syntax**:- $passwd

**Explanation**:-Changes the password

**Command**:- echo

**Syntax**:-$echo googmorning [enter]

**output:**- goodmorning

**Explanation**:-used to display message

**Command**:- date

**Syntax**:- $date

**Explanation**:-display the current date along with the time nearest to the second.

**date options**:-

**Option**: - m

**Syntax**: - $date +%m

**output**: 09

**Explanation**:-display only the month in the numeric form

**Option**: - h

**Syntax**: -$date +%h

**output**: Sep

**Explanation**:-display only the month in the sting form

**Option**: - h&m

**Syntax**: -$date +”%h %m”

**output**: 09 Sep

**Explanation**:-display only the month in the sting form

**Option**: -u

**Syntax**: -$date -u

**output**: Sat Sep 25 05:58:20 UTC 2004

**Explanation**:-displays the universal time

(Greenwich Mean Time)

**Command**:- cal [calendar]

**Syntax**:- $cal

**Explanation**:- It print the calendar of a specific month or a specific year

**example**: -cal 09 1949

**Command**:- bc [basic calculator]

**Syntax**:-$bc [enter]

sqrt 55

7

quit

**Explanation**: - bc command is both a calculator and a small language for writing numerical programs.

**File commands**

**Command**: -mkdir

**Syntax**: - $mkdir <dirname>

**Explanation**:- Creates a directory with the specified name.

**Example**: - $mkdir dir1

**Command**: -cd

**Syntax**: - $cd <dirname>

**Explanation**:- Changes to the specified directory.

**Example**: - $cd dir1

**Command**: - rmdir

**Syntax**: - $rmdir <dirname>

**Explanation**:- Deletes the empty directory.

**Example**: - $rmdir dir1

**Command**: - rm

**Syntax**: - $rm <filename>

**Explanation**:- Deletes a file.

**Example**: - $rm file1

**rm options**:-

**Option**: - -R

**Syntax**: - $rm -R <dirname>

**Explanation**:- Deletes a directory which is not empty.

**Example**: - $rm -R dir2

**Option**: - -i

**Syntax**: - $rm -i <filename>

**Explanation**:- Deletes a file interactively.

**Example**: - $rm -i file2

**Option**: - -F

**Syntax**: - $rm -F <filename>

**Explanation**:- Deletes a file forcibly.

**Example**: - $rm -F file3

**Command**: -touch

**Syntax**: - $touch <filename1> <filename2> ------ <filenamen>

**Explanation**:- Creates n empty files with 0 bytes each

**Example**: - $touch file1 file2 file3

**Command**: -cat

**Syntax**: - $cat > <filename>

**Explanation**:- Creates a file with specified content.

**Example**: - $cat > file1

Hello this is my new file.

^d

**Syntax**: - $cat <filename>

**Explanation**:- Displays the contents of the file specified.

**Example**: - $cat file1

Hello this is my new file.

**File permissions commands**

**Command**:- chmod

**Syntax**:- $chmod [who] [+/-/=] [permissions] <filename>

**Explanation:-** UNIX by default gives permissions for a file and for a directory. We can change these permissions using chmod command.

Here who can be u (for owner)

g (for group)

o (for others)

a (for all)

+/-/= can be + (for adding permission)

- (for removing permission)

= (instructs chmod to add the specified permissions and take away all others, if present)

Permissions can be

r (for read)

w (for write)

x (for execute)

**Examples**:-

1. $chmod +w myfile

Gives write permissions to all.

2. $chmod go-x myfile

Takes away execute permission from groups as well as others.

3. $chmod go+r,go-w myfile

Gives read permission to group and others and take away write permission for a file

4. $chmod go=r, u=rw myfile

Removes all existing permissions and replaces them with read permission for group and others and read and write permissions for owner of myfile

Instead of using u/g/o and +/-/= we can use the weights read(4) write(2) execute(1).

5. $chmod 744 file1

Assigns the permissions rwxr--r--

6. $chmod 777 file1

Assigns the permissions rwxrwxrwx

7. $chmod 654 file1

Assigns the permissions rw-r-xr--

8. $chmod 457 file1

Assigns the permissions r--r-xrwx

**Command**:- ulimit

**Syntax**:- $ulimit

**Explanation**:- ulimit stands for user limit and contains a value which signifies the largest file that can be created by the user in the file system

**Displaying the contents of a File commands**

**Command**: -ls

**Syntax**: - $ ls

**Explanation**:-It displays list of files

**ls options**:-

**Option**: - -l

**Syntax**: - $ls -l

**Explanation**:- displays long listing.

**Option**: - -t

**Syntax**: - $ls -t

**Explanation**:- sorts according to time of modification.

**Option**: - -a

**Syntax**: - $ls -a

**Explanation**:- includes hidden files.

**Option**: - -alR

**Syntax**: - $ls -alR

**Explanation**:- displays all the directories, files, subdirectories and subdirectory files.

**Option**: - -alt

**Syntax**: - $ls -alt

**Explanation**:- displays by combining the properties of a, l and t.

**Command**: - head

**Syntax**: - $head <filename>

**Explanation**:- displays the lines in the beginning of the file

**Example** 1:- $head file1

**Explanation**:- displays first 10 lines of file named file1.

**Example** 2:- $head -3 file2

**Explanation**: - displays the first 3 lines of the file named file2.

**Example** 3:- $head -c47 file3

**Explanation**:- displays up to 47 characters in the file named file3.

**Command**: - tail

**Syntax**: - $tail <filename>

**Explanation**:- displays lines at the end of the file.

**Example** 1:- $tail file1

**Explanation**:- displays last 10 lines of file named file1.

**Example** 2:- $tail -3 file2

**Explanation**:- displays the last 3 lines of the file named file2.

**Example** 3:- $tail -512c file3

**Explanation**:- displays the last 512 characters in the file named file3.

**Command**:- nl

**Syntax**:- $nl <filename>

**Explanation**:-displays the contents of the file along with the line numbers.

**Example**:- $nl file1

**Command**:- uniq

**Syntax**:- $ uniq <filename>

**Explanation**:-It deletes the duplicate entries in a file. It requires sorted file as input.

**uniq options**:-

**Option**:- -u

**Syntax**:- $uniq -u <filename>

**Explanation**:-Selects only non repeated lines in file.

**Example**:- $uniq -u filexx

**Option**:- -d

**Syntax**:- $uniq -d <filename>

**Explanation**:-Selects only repeated lines in file.

**Example**:- $uniq -d filexx

**Option**:- -c

**Syntax**:- $uniq -c <filename>

**Explanation**:-Counts frequency of occurrences of lines in file.

**Example**:- $uniq -c filexx

**Command**:- cut

**Explanation**:-Slits file vertically

**cut options**:-

**Option**:- -c

**Example**:- $cut –c 1-4 filex

**Explanation**:-Displays the first four characters of each line in filex

**Option**:- -f

**Example**:- $cut -f 1,3 filea

**Explanation**:-Displays the fields 1, 3 in filea where the delimiter is tab.

**Option**:- -d

**Example**:- $cut -d**:** -f 1,3 filess

**Explanation**:-Displays the fields 1, 3 in filess where the delimiter is **:**

**Command**:- paste

**Syntax**:- $paste <filename1> <filename2>

**Explanation**:-What we cut with cut command can be pasted back with paste command.

**Example**:- $paste file1 file2

**Command**:- move

**Syntax**:-$ mv [options] source destination

**Explanation**: mv cmd is used to move or rename files and directories.

**Examples**:-

**Options :**

1)$mv file1 file2 #renaming a file

2) Use -i(–interactive)

[Before overwriting existing destination file prompt for confirmation.]

$ mv -i test1 test2 oldbackup/

mv: overwrite `oldbackup/test1'? y

mv: overwrite `oldbackup/test2'? y

3) Use –f (–force)

[ Forcefully overwrite existing destination file. ]

$ mv -f test1 test2 oldbackup/

4) Use -n(–no-clobber)

[Existing file will not overwrite]

$ mv -n test1 oldbackup/

**Command**:-copy

**Syntax**:- $ cp [options] source destination

**Explanation**: used to copy data from one files another and directories also.

**Examples**:-

1) $cp file1 file2

[Copy the contents of file1 to file2. If file2 does not exist, it will be created.]

2) $ cp file1 file2 Files/

[Copy multiple files in single directory. Provided that directory exists.]

**Command**:-wc [wordcount]

**Syntax**:- $ wc [options] filenames

**Explanation**: used to find out number of newline count, word count, byte and characters count in a files specified by the file arguments.

**wc options**:-

wc -l : Prints the number of lines in a file.

wc -w : prints the number of words in a file.

wc -c : Displays the count of bytes in a file.

wc -m : prints the count of characters from a file.

wc -L : prints only the length of the longest line in a file.

**Examples:**

1) $ wc tecmint.txt

12 16 112 tecmint.txt

2) $ wc -l tecmint.txt

12 tecmint.txt

3) $ wc -w tecmint.txt

16 tecmint.txt

4) $ wc -c tecmint.txt

112 tecmint.txt

5) $ wc -m tecmint.txt

112 tecmint.txt

6) $ wc -L tecmint.txt

16 tecmint.txt

**Command**:- join **Syntax**:- $join <filename1> <filename2>

**Explanation**:-Joins the lines which have the first word(field) in common. Join takes first word for comparing by default.

**Example**:- $cat hello

Hello how are u

$cat hi

Hello where are u

$join hello hi

Hello how are u where are u

**Join options**:-

**Option**:- -j

**Syntax**:- $join –j <fieldno> <filename1><filename2>

**Explanation**:- If we want to compare the word other than the first word then we have to use this option followed by the field number. The common field word will be placed first followed by the contents of the files one and two respectively.

**Example** 1:- $join -j 3 hello hi

Are hello how u hello where u

**Networking command**

**Command**:- hostname

**Syntax**:- $ hostname

**Explanation**:- finding host/domain name and IP address

**options:**

hostname with no options displays the machine's hostname

* hostname –d displays the domain name the machine belongs to
* hostname –f displays the fully qualified host and domain name
* hostname –i displays the IP address for the current machine

**Command**:- netstat

**Syntax**:- $ netstat

**Explanation**:- Network connections, routing tables, interface statistics –

Most useful and very versatile for finding a connection to and from the host. You can find out all the multicast groups (network) subscribed by this host by issuing "netstat -g"

**options:**

netstat -nap | grep port will display process id of application which is using that port

netstat -a or netstat –all will display all connections including TCP and UDP

netstat --tcp or netstat –t will display only TCP connection

netstat --udp or netstat –u will display only UDP connection

netstat -g will display all multicast network subscribed by this host.

**Command**:- nslookup

**Syntax**:- $ nslookup blogger.com

**Explanation** If you know the IP address it will display hostname. To find all the IP addresses for a given domain name, the command nslookup is used. You must have a connection to the internet for this utility to be use.

**Command**:- traceroute

**Syntax**:- $

**Explanation** A handy utility to view the number of hops and response time to get to a remote system or website is traceroute. Again you need an internet connection to make use of this tool.

**Command**:- finger

**Syntax**:- $

**Explanation** View user information, displays a user’s login name, real name, terminal name and write status,rarely used nowadays.

**Command**:- telnet

**Syntax**:- $ telnet hostname port

**Explanation** Connects destination host via the telnet protocol, if telnet connection establishes on any port means connectivity between two hosts is working fine. It will telnet hostname with the port specified. Normally it is used to see whether the host is alive and the network connection is fine or not.

**REGULAR EXPRESSIONS**

# grep command in 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 **globally search for regular expression** and printout).

**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.

**Sample Commands**

Consider the below file as an input.

**$cat > geekfile.txt**

unix is great os. unix is opensource. unix is free os.

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 give file. It matches the words like “UNIX”, “Unix”, “unix”.

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

**Output:**

unix is great os. unix is opensource. unix is free os.

Unix linux which one you choose.

uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

1. **Displaying the count of number of matches :**

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

**$grep -c "unix" geekfile.txt**

**Output:**

1. **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

1. **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 "unix" geekfile.txt**

**Output:**

unix is great os. unix is opensource. unix is free os.

uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

1. **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 opensource. 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 sting 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 opensource. 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

**examples:**

**Command**:- grep

**Syntax**:- $grep options pattern filename(s)

**Explanation**:-Searches for the pattern in files and displays the matching lines.

**Example** 1:- $grep sales emp.lst

Displays lines which contain the word “sales” in the file emp.lst

**Example** 2:- $grep director emp1.lst emp2.lst

Displays lines which contain the word “director” in the two files.

**Command**:- egrep (**extended grep**)

**Example**:- $egrep ‘woodhouse | woodcock’ emp.list

**Explanation**:-Displays lines which contain either woodhouse or woodcock.

**Command**:- fgrep (**fixed character grep**)

**Example**:- $fgrep -f ‘woodhouse|woodcock’ emp.lst

**Explanation**:-displays lines which contain either woodhouse or woodcock

**VI EDITOR**

**VI EDITOR** is treated as one of the popular editor in **UNIX** since last 30 years.

It has three modes.

1. Input mode
2. Command mode
3. Colon mode

* Input mode in which what ever user types will be written into the document.
* Command mode is the one in which user can enter commands.
* Colon mode in which users can run commands to do some editing on the document content.

The following commands are summarized to work under **UNIX**.

1. **vi** filename - Opens **vi** editor with the given filename.

2. When the editor opens, a screen will be opened with the command mode.

3. To enter text press **i** then input mode will be displayed at bottom right part of the screen.

4. To stop typing press **Esc** key then command mode comes. Press

**: w** to save the matter and resume editing.

**: wq** to save the matter and quit the vi editor.

**: q!** to quit the editor without saving.

5. In command mode commands can be entered. By pressing **Esc** key one can go to command mode from other modes.

1. Press **i** to insert text before the current cursor position.
2. Press **I** to insert text at the beginning of the line.
3. Press **a** to insert text after the cursor position.
4. Press **A** to insert text at the end of the current line.
5. Press **o** to open a new line below the current line.
6. Press **O** to open a new line above the current line.
7. Press **r** to replace the present character with a character.
8. Press **R** to replace a group of characters from current cursor position.
9. Press **x** to delete present character.
10. Press **J** to join the next line to the end of the current line.
11. Press **dd** to delete the current line.
12. Press **4dd** to delete 4 lines from the current line.
13. Press **dw** to delete the current word.
14. Press **7dw** to delete 7 words from the current word onwards.
15. Press **30i\*Esc** to insert 30 \*’s at the current position.
16. Press **u** to undo effect the previous command on the document.
17. Press **.** to repeat the previous command on the document.
18. Press **yy** to copy the entire line in to the buffer.
19. Press **yw** to copy the entire word in to the buffer.
20. Press **p** to place the copied or deleted information below the cursor.
21. Press **P** to place the copied or deleted information above the cursor.

Colon mode commands

1. Search and Substitute commands
2. **:/raja**

searches for the string ‘raja’ in the forward direction.

Press n to repeat the search.

1. **:?raja**

search for the string in the backward direction.

Press n to repeat the search.

1. **:s/raja/rama**

replaces the first occurrence of “raja” with “rama”.

1. **:s/raja/rama/g**

replaces all “raja” ‘s with “rama” in the present line.

5.**:g/raja/s/raja/rama/g**

replaces all “raja” ‘s with “rama” in whole file.

1. Block delete commands

1. **:1d** deletes the line 1.

2. **:1,5d** deletes the lines from 1 to 5.

***NOTE***: - $ means last line of the file.

. means present line.

3. **:10,$d** deletes lines from 10th line to the last line of the file.

4. **:1,$d** deletes lines from 1st line to the last line of the file.

5. **:.,$d** deletes lines from present line to the last line of the file.

6. **:.-3,.d** deletes the lines from present line and above 2 lines.