TR Commands :

1. To delete a char: $cat filename.txt | tr -d ‘a’

​​ This option deletes characters in the set specified.

2. To remove digits from a string: $cat filename.txt | tr -d [:digit:]

​​​This command deletes the numbers from the given string.

​​​​ $cat filename.txt | tr -cd [:digit:]

​​​This command removes all characters except digits.

3. To find the hexa-decimal(a-z, A-Z 0-9): $cat filename.txt | tr -d [:xdigit:]

​​​This command eliminates the hexadecimal values.

​​​​ $cat filename.txt | tr -cd [:xdigit:]

​​​This command removes all characters except hexa-decimal digits.

4. To remove character from a string: $cat filename.txt | tr -d [:alpha:]

​​​This command deletes all the characters from the given string.

​​​​ $cat filename.txt | tr -cd [:alpha:]

​​​This command removes all digits and parenthesis except characters.

5. To replace character with some other: $ echo “this is a string ” | tr ‘ia’ ‘14’

o/p-->(th1s 1s 4 str1ng)

6. To discard unwanted spaces: $ echo “this is​a string” | tr -s ‘ ’ ‘ ’

​​​ ​ o/p--> this is a string

7. To change new line character with space: $tr -s ‘\n’ ‘ ’ < file.txt

​ This command will change new line character with space.

8. To change {} to (): $tr ‘{}’ ‘()’ <inputfile> outputfile

​This command will change {} to ()

9. To convert lower case to upper case: $cat filename.txt | tr “[a-z]” “[A-Z]”

$cat filename.txt | tr “[:lower :]” “[:upper :]”

This command converts lower case to upper case.

10. To Translate white-spaces to tabs: $ cat filename.txt | tr [:space:]

This Command translate white-spaces in a string to tabs.

11. To Translate braces to parenthesis: $cat filename.txt | tr “{ }” “( )”

$cat filename.txt | tr “hello” “hiiii” (change the characters from hello to hiiii i.e.,h-h,e-i,l-i,l-i,o-i)

These two commands helps in translating the characters and the special symbols.

To eliminate repeated digits or characters

\* awk ‘!a[$0]++’

[$0] – represents current line

It will be stored in an array ‘a’

‘++’ - increment

\* cat file.txt | awk ‘!a[$0]++’

Example: cat > file.txt cat>num.txt

​​ram 123

​​ravan 234

​​laxman 123

​​ram 456

​​arun 123

​​ajay

​​ram

​​

​​cat file.txt | awk ‘!a[$0]++’ cat num.txt | awk ‘!a[$0]++’

​​ram 123

​​ravan 234

​​laxman 456

​​arun

​​ajay

Find Command:

Different operations using find command:

1) Find and delete a file with confirmation

$ find ./ -name file\_name -exec rm -i {} \;

When this command is entered, a prompt will come for confirmation, if you want to delete the file or not. If you enter ‘Y/y’ it will delete the file. Instead, if you enter ‘N/n’ it will not delete the file.

Ex: $find /home/user/Public -name new.txt -exec rm -i {} \;

rm: remove regular file ‘/home/user/Public/new.txt’ (If you enter y or Y, it will delete the file)

2) Find and delete a file without confirmation

$ find / -name file\_name -exec rm {} \;

When this command is entered, the file will be deleted directly without asking for any confirmation.

Ex: $find /home/user/Public -name new.txt -exec rm {} \;

Note: The file gets deleted

3) Search for empty files and directories

$ find / -empty

This command find all empty folders and files in the entered directory or sub-directories.

Ex: $find /home/user/Public -empty

/home/user/Public/new.txt

/home/user/Public/new2.txt

/home/user/Public/abc

4) Find all Empty Files

$ find / -type f -empty

This command is used to find all empty files under a certain path.

Ex: $find /home/user/Public -type f -empty

/home/user/Public/new.txt

/home/user/Public/new2.txt

5) Find all Empty Directories

$ find / -type d -empty

This command is used to find all empty directories under a certain path.

Ex: $find /home/user/Public -type d -empty

/home/user/Public/abc

6) Search text within multiple files

$ find / -type f -name "\*.txt" -exec grep 'any\_word' {} \;

This command print lines which have the word specified in them and ‘-type f’ specifies the input type is a file.

Ex: $ find /home/user/Public -type f -name "\*.txt" -exec grep 'cat' {} \;

I’m a cat.

cat

7. Tree command – This command helps you list all files and directories under a specified directory.

$ tree -a directory\_name

8. Find a specified file – This command shows the path where the specified file is present

$ find directory\_name -name file\_name.txt

9. Find a list of files - This command shows the path where a list of files is present

$ find directory\_name -name “\*.txt”

Finds all the text files under a particular directory

$ find directory\_name -name “\*t.txt”

Finds all the text files ending with “t” under a particular directory

$ find directory\_name -name “t\*.txt”

Finds all the text files starting with “t” under a particular directory

10. Find list of files having specified permissions – Displays all the files under the specified directory having the specified permission

$ find directory\_name -perm 664

COPY Commands

cd pavan

cat>ss1.txt

hello

hello good moring

cd ~

mkdir torris

cd pavan

cp ss1.txt/home/user/torris

cd ~

cd torris

ls

cat ss1.txt

hello

hello good moring

mv ss1.txt ss2.txt

cat ss2.txt

hello hiii

hello good moring

cp -i a.txt b.txt

Did u want to overwrite? Y

Content of the a.txt is overwrite to the b.txt the content of the b.txt will be removed.

cp -b a.txt b.txt

Content of the a.txt is overwrite to the b.txt and content of the b.txt is backups as b.txt~

cp a.txt b.txt dir name

The both txt files should copied to the new directory.

cp \*.txt dir name

The all .txt files in the one directory is copied to the other directory.

cp -f a.txt b.txt dir name

It force fully copied to the other directory.

MOVE Commands

mv file1 file2

mv pavan/file1 Harris/

mv pavan/file1 Harris/file2

mv pavan/\* Harris/

mv -u pavan/\* Harris/

mv —suffix=.txt pavan/file1 Harris/file2

mv -i pavan/file1 Harris/file2

LS Commands

ls -g :- donot list the owner

ls -G :- in a long listing dont print group names

ls -H :- follow symbolic links listed on the command line

ls -i :- prints the index no. of file

ls -L :- shows info of the file

ls -m :- fill width with a comma separated list of entries

ls -n :- list numeric user and group ids

ls -N :- print raw entry names

ls -o :- do not list group info

COPY Commands

Cut -b 1,2,3 file.txt = List without ranges

cut -b 1-3,5-7 state.txt = List with ranges

cut -b 1- state.txt = from 1st byte to end byte

cut -b -3 state.txt = from 1st byte to 3rd byte

cut -c4 file.txt = to print characters in a line by specifying the position of the characters.

cut -c4,6 file.txt = more than one character at a time

cut -c4-7 file.txt = print a range of characters in a line by specifying the start and end position of the characters.

cut -c-6 file.txt = first six characters in a line, omit the start position and specify only the end position.

cut -c10- file.txt = To print the characters from tenth position to the end, specify only the start position and omit the end position.

cut -c- file.txt = If you omit the start and end positions, then the cut command prints the entire line.

cut --complement -c 5 state.txt = except 5th char all will be there

SORT Commands

Sort file.txt-Sort the content in ascending order

Sort –r file.txt-display the sort in reverse

Sort –k 2 file.txt-Sort with respect to 2nd field

Sort –c file.txt-check the content in the file is sorted

Sort –u file.txt-sort and removes duplicate

Sort –n file.txt-sort the number

Sort –nu file.txt-sort the number and removes duplicate

Sort –M file.txt-sort the months

Sort –o output.txt file.txt-sorts and saves the output in output file

Sort –t “,” –k 1 file.txt-sort in 1st field for delimiter in the content

Sort –t “,” –k2nr file.txt-sort in 2st field for delimiter in the content and reverse it

LESS Commands

View a text file with less

As showed in the syntax, you can use the less command to view a file in the following fashion:

less [option]<filename>

Exit from less

If you are not used to of less command, you might struggle to find how to exit less. Trust me it’s not at all complicated. Simply press ‘q’ at any given point to exit from less.

Moving around in less

The output of less is divided into sort of pages. You’ll see only the text that fills up to your terminal screen.

Up arrow – Move one line up

Down arrow – Move one line down Space or PgDn – Move one page down b or PgUp – Move one page up

g – Move to the beginning of the file G–Movetotheendofthefile

ng – Move to the nth line

Display line numbers with less

If you want to see the line numbers in the less command output, you can use the option N in the following manner:

less -N <filename>

Finding text in less

If you have a large text file, it’s better to search for a specific piece of text rather than reading it line by line in order to find it manually.

To find a word or phrase or even a regex pattern, press / and type whatever you want to find.

/pattern

Viewmultiplefileswithlesscommand

To open multiple files with less, simply input the file names one by one:

less <filename1> <filename2> <filename3> :n to move next file

:p to move to previous file

less -X file name

save the file name on screen when file exits

less -S file name

long line can be seen by side wrapping.

less--version toknowtheversionofless 10. less -m file name

show more detailed prompt including file position.

HEAD Commands

Headcommandwithfourtags

-n number of lines -cBytes -qquiet -vverbose

By default :

$ head file.txt

$ head file.txt file2txt file3.txt of the given input.

The head command, as the name implies, print the top N number of data . By default, it prints the first 10 lines of the specified files. If more than

one file name is provided then data from each file is preceded by its file name.

-nnumber of lines

$head -n 5 file.txt ---It prints top 5 row in the file.

$head -n -5 file.txt ---It prints all rows except last 5 row in the file.

$head -n 0 file.txt ---It prints nothing. Because we gave 0 lines. Hence it

shows nothing.

$head -n -0 file.txt ---It prints all whatever the file consist of txt.(inverse of -

previous command)

-cbytes

$head -c 5 file.txt ---It prints top 5 bytes of character in the file.

$head -c -5 file.txt ---It prints all rows except last 5 character in the file.

Other combinations:

$ls /etc | head -n 15 what the ls prints from that it will prints top 15 lines.

$ls | head -c 5what the ls prints from that it will print initial 5 character .

$ls | head -c -5 what the ls prints from that it will prints all except last 5

character.

$head file.txt | sort whatever the file.txt consist . It will print in sorted list.

$head file.txt | sort -r-- whatever the file.txt consist . It will print in sort in

reverse order.

$head –version -It displays the version of head command.

$head -n 10 file.txt | tail -5-It prints line from 6 to 10 .

pwd---->present

working directory

Head

ls---->shows the files in directory

cat f1.txt f2.txt----> display the contents of f1.txt and f2.txt

head -v f1.txt---->display the file name along with contents

head -q f1.txt f2.txt---->-q is used for more than one files

head -q -n 4 f1.txt f2.txt

head -q -n -4 f1.txt f2.txt

head -q -c 4 f1.txt f2.txt

head -q -c -4 f1.txt f2.txt

cd~ ls

ls -t

ls -t | head -n 3 |sort

compgen -a | head -n 2

TAIL Command

Let us consider two files

1) states.txt

2) capitals.txt

$tail states.txt // this command will print last 10 lines of specified file (default).

-n num : It prints the last num lines.

$tail -n 3 states.txt // this will print last 3 lines of specified file. $tail -3 states.txt // this is also same as above command instead of

using -n we can give “-“ sign infront of number. $tail+20states.txt //thiswillprintstartingfromline20.

-c num: It prints last num bytes from the file specified.

$tail -c -6 states.txt // this will print last 6 characters of specified file.

-q: It is used if more than 1 file is given.

files.

-v: By using this option, data from the specified file is always preceded

It is the complementary of head command. The tail command, as the name

implies, print the last N number of data of the given input.

$tail -q states.txt capitals.txt // this will combine the 2 specified files.

$tail -q -n 1 states.txt capitals.txt // this will give first line of both the

by its file name.

$tail -v states.txt capitals.txt

The tail command can be piped with many other commands of the unix. $ tail -n 7 state.txt | sort -r //

$ cat states.txt | head -n 20 | tail -n 5 > list.txt //First cat

–version:

$ tail –version

This option is used to display the version of tail which is currently running on your system.

output of the tail command is given as input to

the sort command with -r option to sort the last 7 state names coming from file

states.txt in the reverse order.

all the data present in the file states.txt and after that pipe transfers all the

command gives

output coming from

cat

command to the

head

command. Head command

gives all the data from start(line number 1) to the line number 20 and pipe

transfer all the output coming from

head

command gives last 5 lines of the data and the output goes to the file name

command to

tail

list.txt via directive operator.