**COPY COMMAND**

mkdir new

cd new

cat>ss1.txt

hello hii

hello good moring

cd ~

mkdir new

cd new

cp ss1.txt/home/user/new

cd ~

cd shiva

 ls

cat ss1.txt

hello hii

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 COMMAND:**

mv file1 file2

mv pavan/file1 Harris/

mv pavan/file1 Harris/file2

mv pavan/\* Harris/

mv -u pavan/\* Harris/

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

mv -i pavan/file1 Harris/file2

mv command is used to move files and directories

$ mv main.txt def.txt /home/user/rapid/

Options:

1) -i(interactive): option makes the command ask the user for confirmation before moving a file that would overwrite an existing file, you have to press y for confirm moving, any other key leaves the file as it is

$ mv -i a.txt b.txt

mv: overwrite 'b.txt'? y

2) -f(force):

The -f option overrides this minor protection and overwrites the destination file forcefully and deletes the source file

$ ls -l b.txt

-r--r--r--+ 1 User User 21 Sep  7 13:37 b.txt

$ mv demo.txt b.txt

mv: replace 'b.txt', overriding mode 0444 (r--r--r--)? n

$ ls

b.txt  c.txt  d.txt  demo.txt

$ mv -f demo.txt b.txt

$ ls

b.txt  c.txt  d.txt

3) -n (no-clobber): With -n option, mv prevent an existing file from being overwritten.

$ ls

b.txt  c.txt  d.txt  demo.txt

$ cat b.txt

Hello world

$ mv -n demo.txt b.txt

$ ls

b.txt  c.txt  d.txt  demo.txt

-b(backup): With this option, it is easier to take a backup of an existing file that will be overwritten as a result of the mv command. This will create a backup file with the tilde character(~) appended to it.

$ ls

b.txt  c.txt  d.txt  demo.txt

$ mv -b demo.txt b.txt

$ ls

b.txt  b.txt~  c.txt  d.txt

$cp file1.txt   file2.txt

$mv file3.txt   file4.txt

Here the difference between cp and mv command is that the source file will be deleted in mv command after, it is moved to destination path but in case cp command it will duplicate the file and source file will not get deleted

**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:] '\t'

                                                              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

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

**$ tree -a directory\_name**

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

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

1. 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

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

**$ find directory\_name -perm 664**