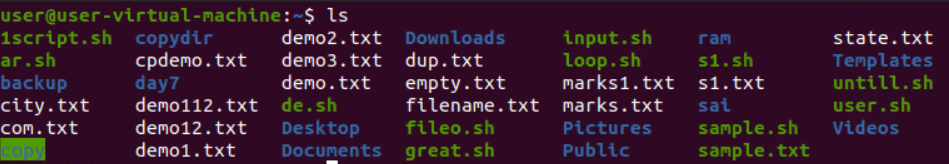
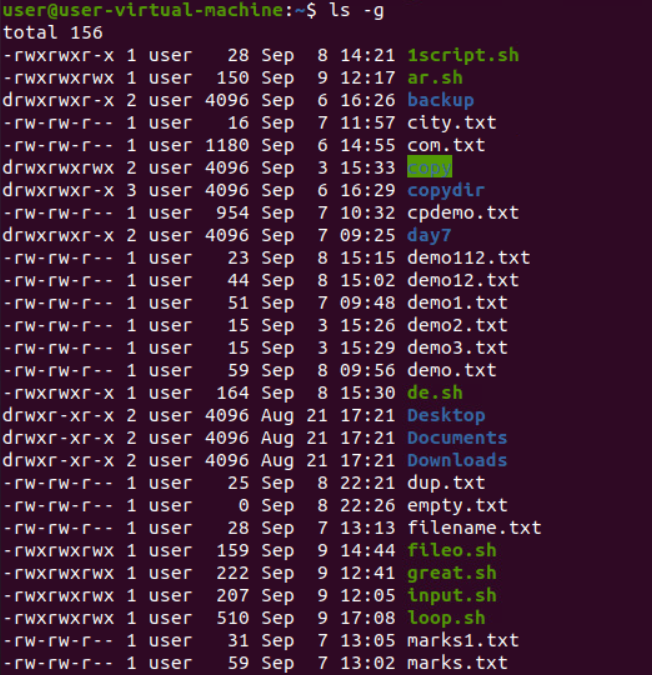
Ls command:

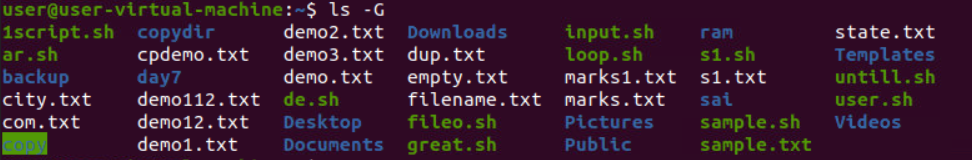
* ls



* ls –g : do not list the owner



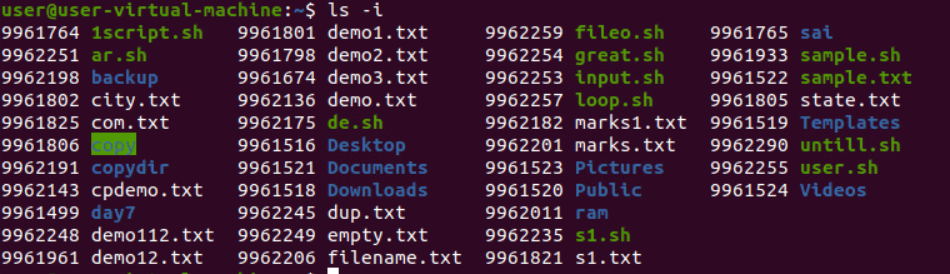
* ls -G : in a long listing don’t print group names



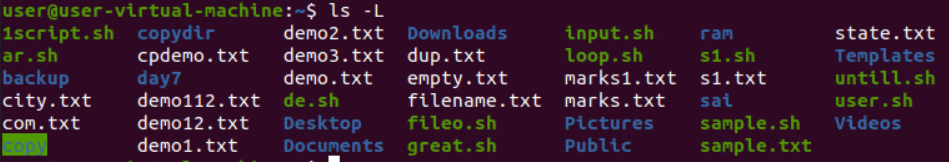
* ls -H : follow symbolic links listed on the command line



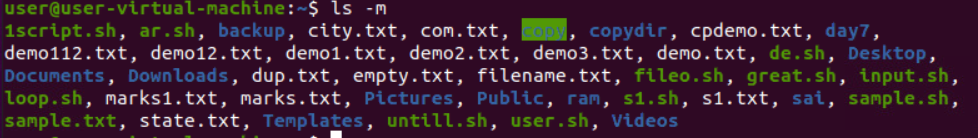
* ls -i : prints the index no. of file



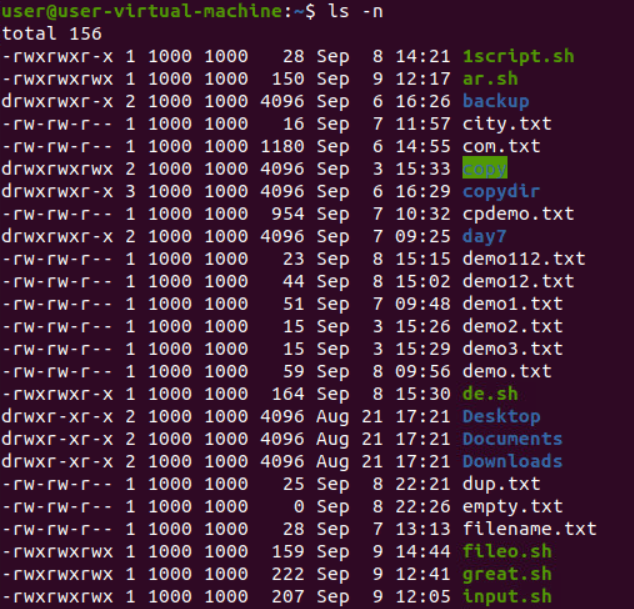
* ls -L : shows info of the file



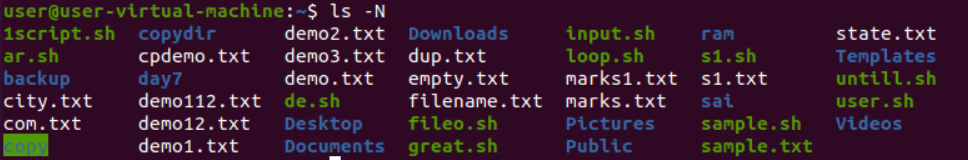
* ls -m : fill width with a comma separated list of entries



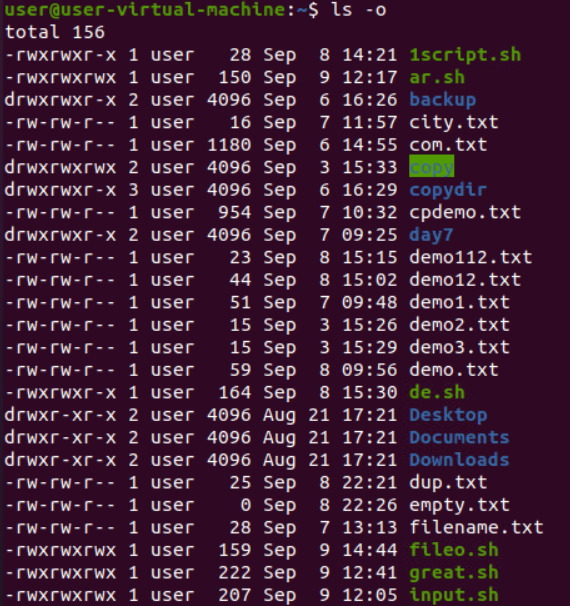
* ls -n : list numeric user and group ids



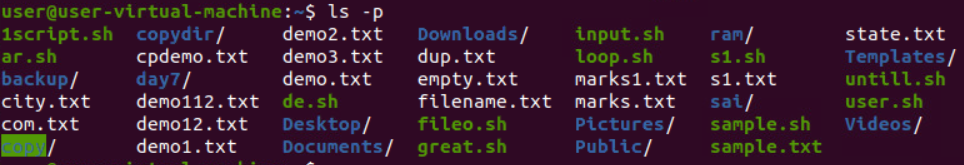
* ls -N : print raw entry names



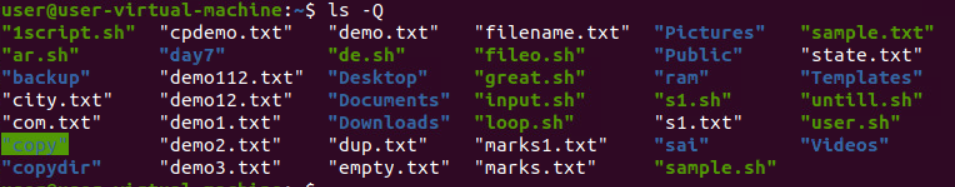
* ls -o : do not list group info



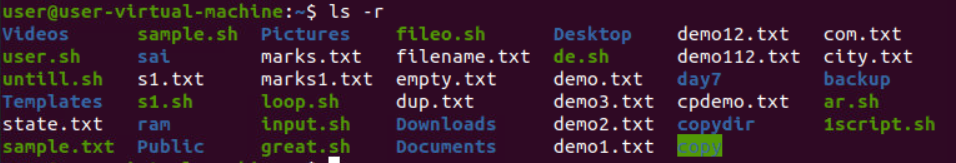
* ls –p : append / indicator to directories(put / in dir file)



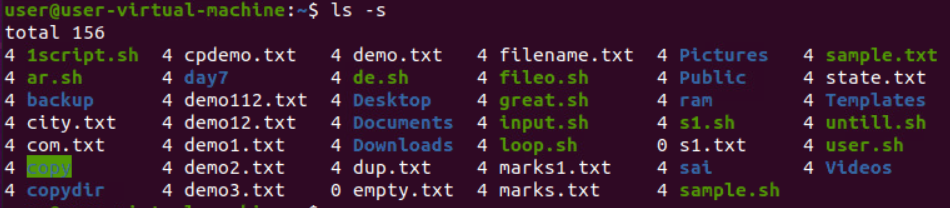
* ls - Q or ls --quote-filename : enclose entry names in double quotes (" ")



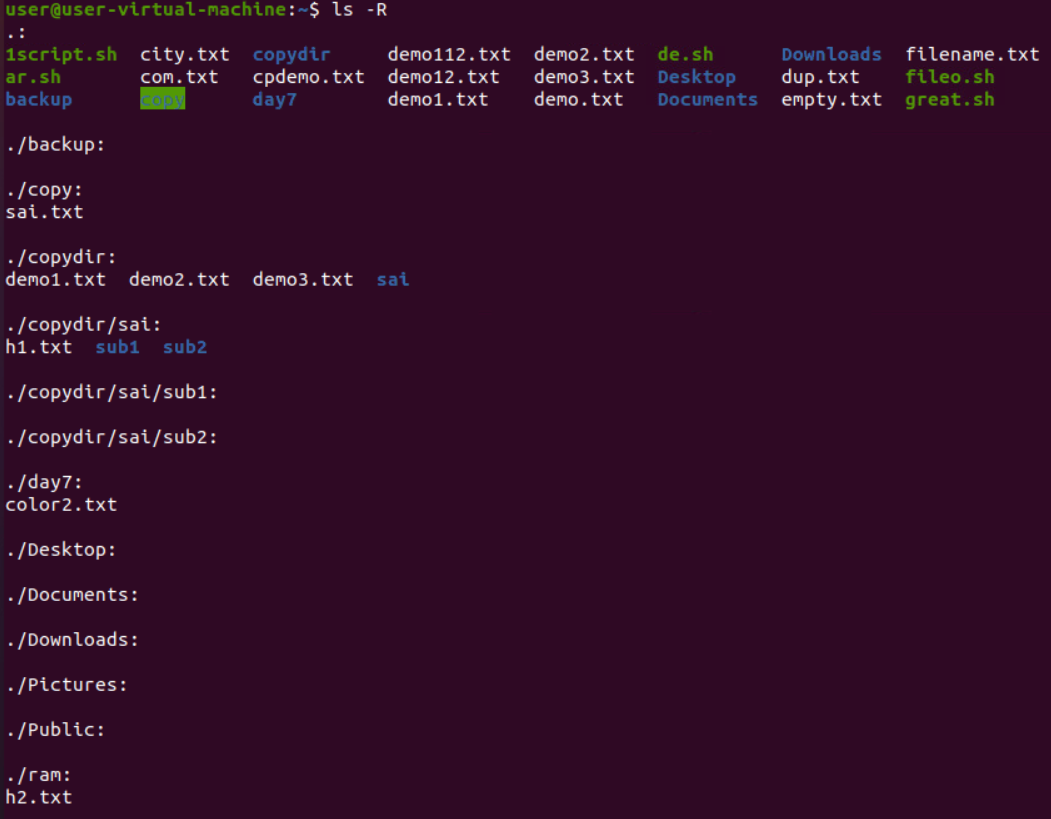
* ls -r or ls –reverse : reverse order while sorting (sorting in reverse order)



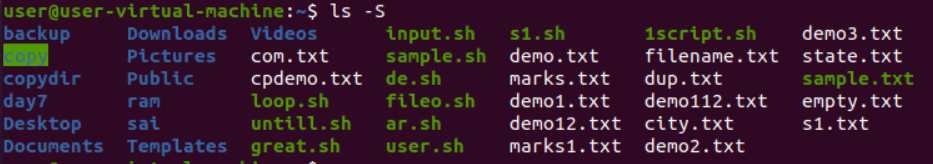
* ls -s or ls –size : print the allocated size of each file, in blocks



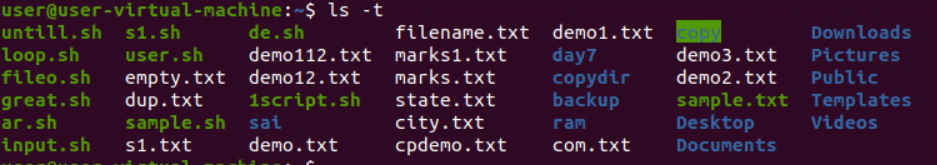
* ls –R : list sub-dirctories recursively



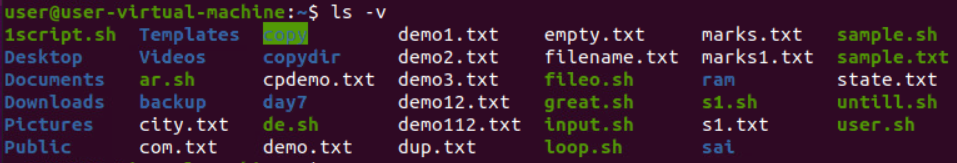
* ls –S : sort by file size



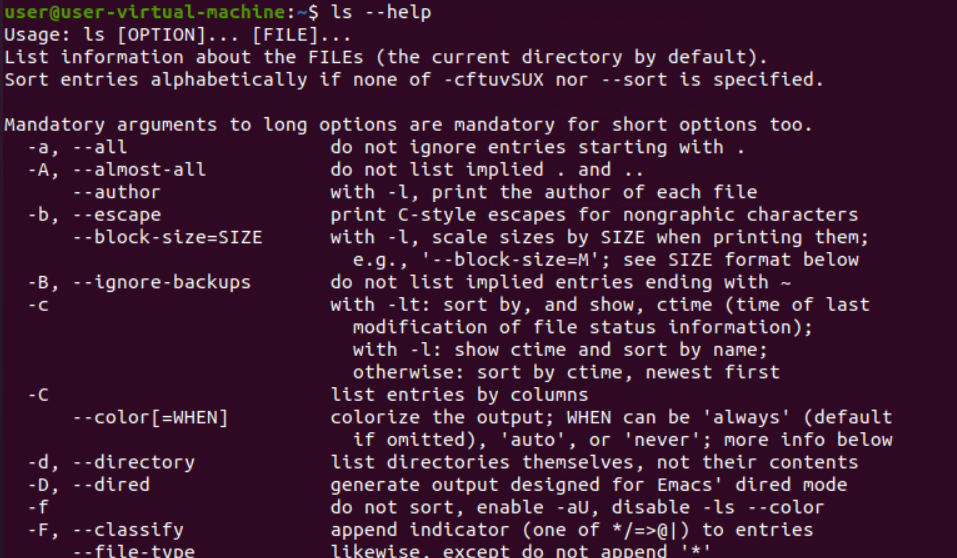
* ls -t : sort by modification time



* ls -v : natural sort numbers within text

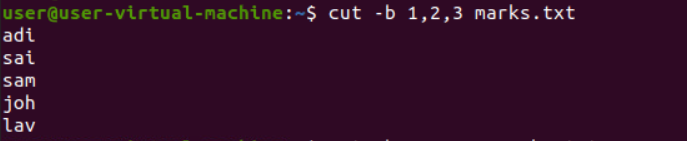


* ls --help : display this help and exit

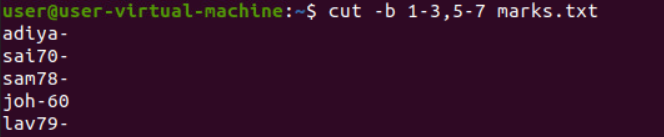


Cut command

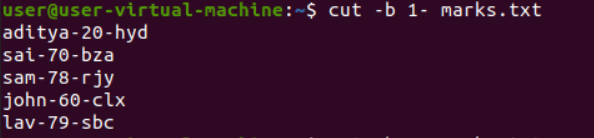
* cut -b 1,2,3 marks.txt : List without ranges



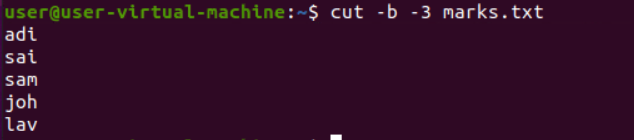
* cut -b 1-3,5-7 marks.txt : List with ranges



* cut -b 1- marks.txt : from 1st byte to end byte.



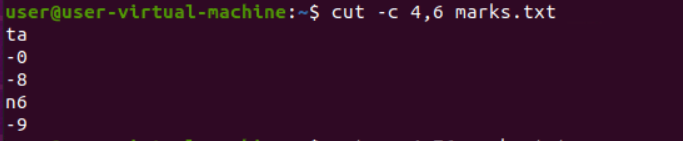
* cut -b -3 state.txt : from 1st byte to 3rd byte



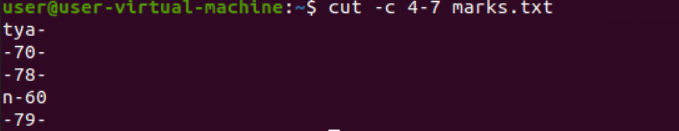
* cut –c 4 marks.txt : to print characters in a line by specifying the position of the characters.



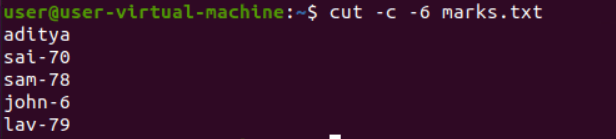
* cut –c 4,6 marks.txt : more than one character at a time.



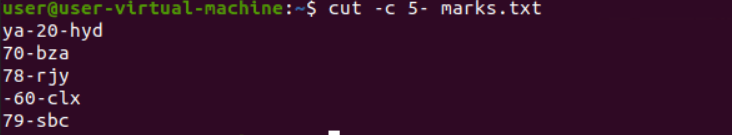
* cut –c 4-7 marks.txt : print a range of characters in a line by specifying the start and end position of the characters.



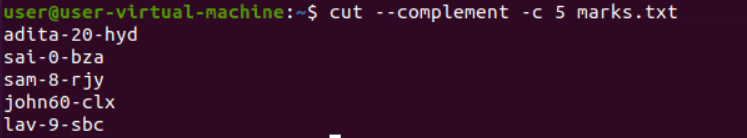
* cut –c -6 marks.txt : first six characters in a line, omit the start position and specify only the end position.



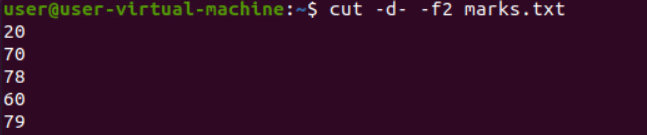
* cut –c 5- marks.txt : To print the characters from tenth position to the end, specify only the start position and omit the end position.



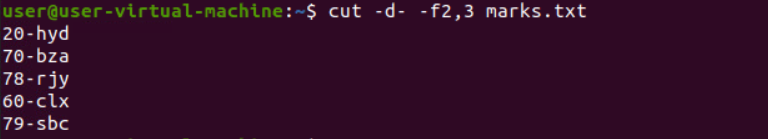
* cut --complement -c 5 marks.txt : except 5th char all will be there.



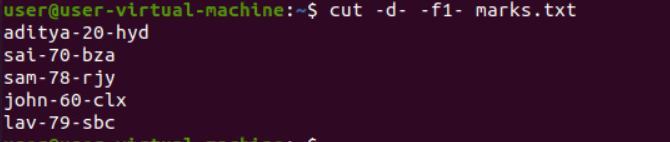
* cut -d- -f2 marks.txt - it shows only the 2nd field.



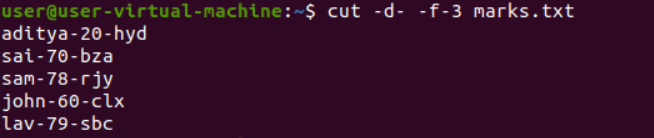
* cut -d- -f2,3 marks.txt - it shows only the 2nd and 3rd fields.



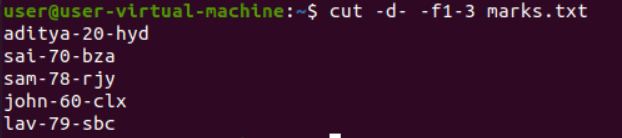
* cut -d- -f1- marks.txt - it shows only fields from 1st field to the last.



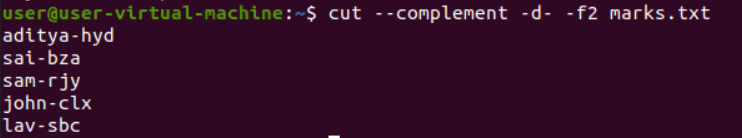
* cut -d- -f-3 marks.txt - it shows only fields from starting field to the 3rd field.



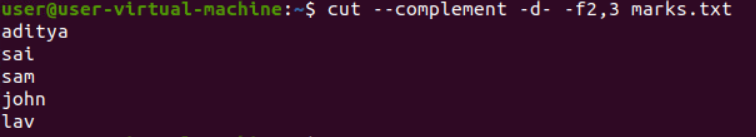
* cut -d- -f1-3 marks.txt - it shows only fields from 1st field to the 3rd field.



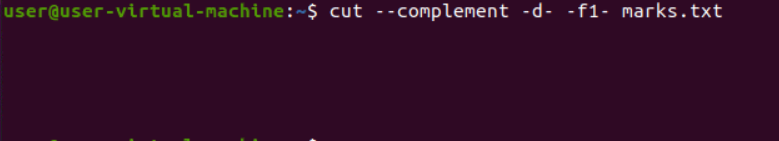
* cut --complement -d- -f2 marks.txt - it shows all the fields except field 2.



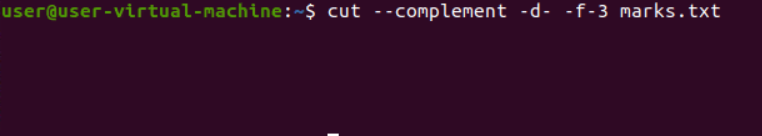
* cut --complement -d- -f2,3 marks.txt - it shows all the fields except field 2 and 3.



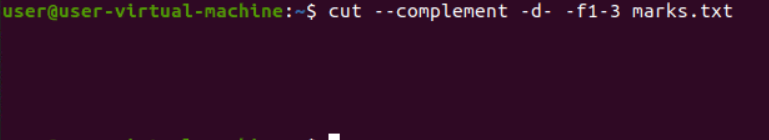
* cut --complement -d- -f1- marks.txt - it does'nt shows the fields from field 1 to the end.



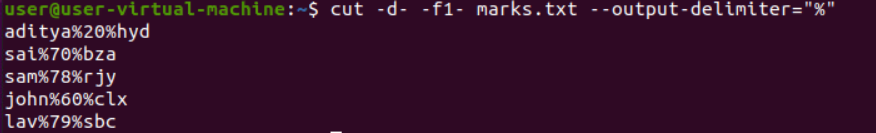
* cut --complement -d- -f-3 marks.txt - it shows all the fields except field from starting field to 3.



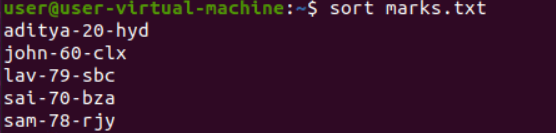
* cut --complement -d- -f1-3 marks.txt - it shows all the fields except field from 1st field to 3rd field.



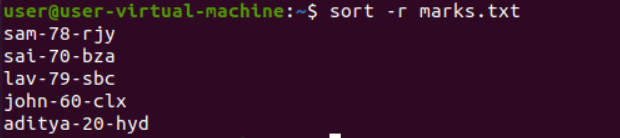
* cut -d- -f2 marks.txt --output-delimiter="%" - it changes the delimiter '-' to '%'



* Sort marks.txt-Sort the content in ascending order.



* Sort –r marks.txt-display the sort in reverse.



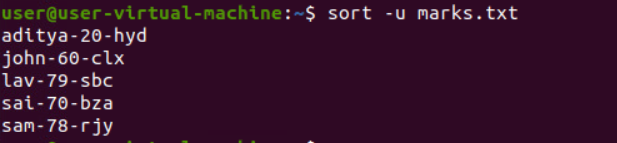
* Sort –k 2 file.txt-Sort with respect to 2nd field



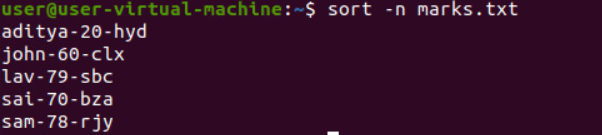
* Sort –c marks.txt-check the content in the file is sorted



* Sort –u marks.txt-sort and removes duplicate.



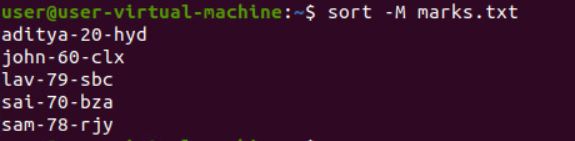
* Sort –n marks.txt-sort the number.



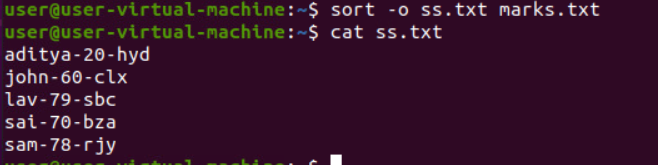
* Sort –nu marks.txt-sort the number and removes duplicate.



* Sort –M marks.txt-sort the months.

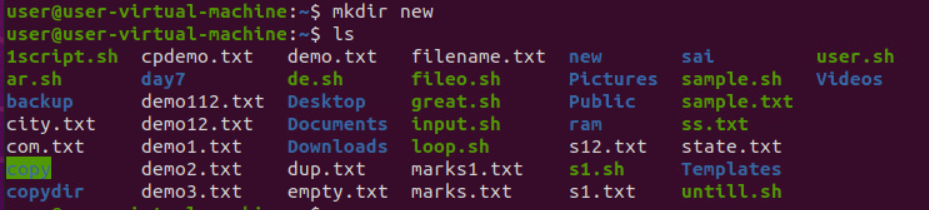


* Sort –o output.txt file.txt-sorts and saves the output in output file.



* Sort –t “,” –k 1 marks.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

**Mkdir**



**Cp command**

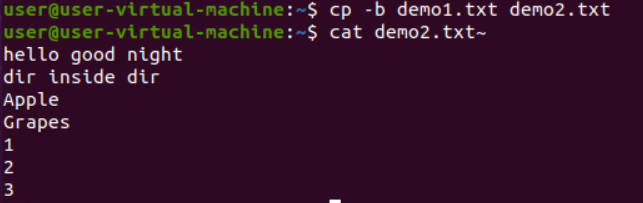
* cp -i demo1.txt demo2.txt

Did u want to overwrite? Y

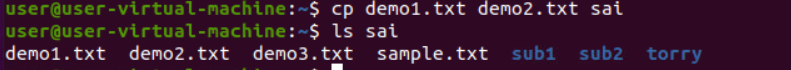
Content of the demo1.txt is overwrite to the demo2.txt the content of the demo2.txt will be removed.



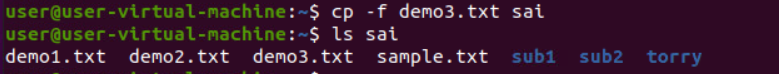
* cp -b demo1.txt demo2.txt : Content of the demo1.txt is overwrite to the b.txt and content of the demo2.txt is backups as demo2.txt~



* cp demo1.txt demo2.txt dir name : The both txt files should copied to the new directory.

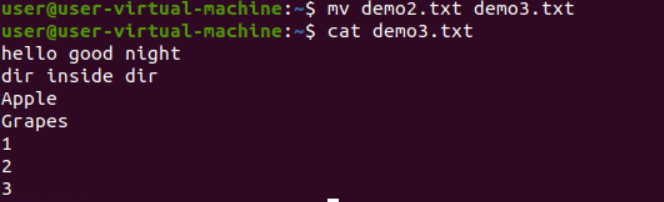


* cp \*.txt dir name : The all .txt files in the one directory is copied to the other directory.
* cp -f demo1.txt demo2.txt dir name : It force fully copied to the other directory.

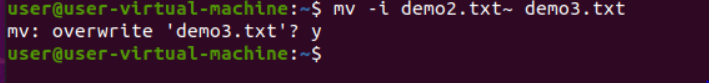


**Mv Command**

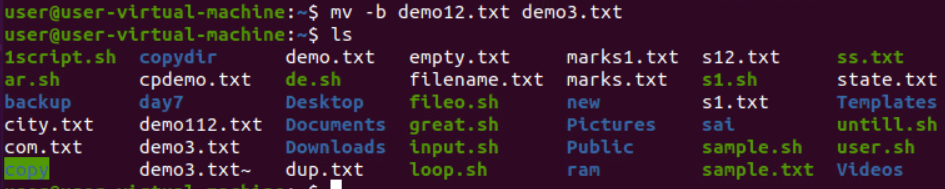
* mv



* mv –i demo2.txt~ demo3.txt : ask users to overwrite file or not.

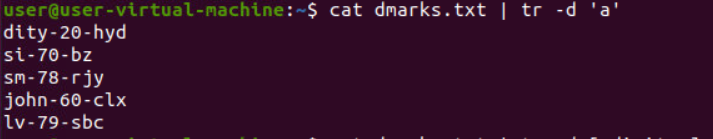


* mv –b demo2.txt demo3.txt : create a backup of an existing file that will be overwritten with the tilde character(~) appended to it. (demo3.txt~)

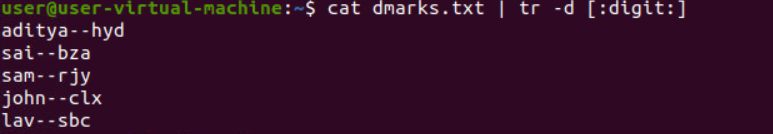


**TR Command**

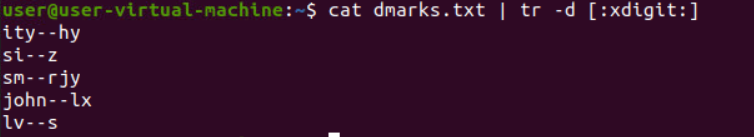
* cat dmarks.txt | tr -d ‘a’ : This option deletes characters in the set specified.



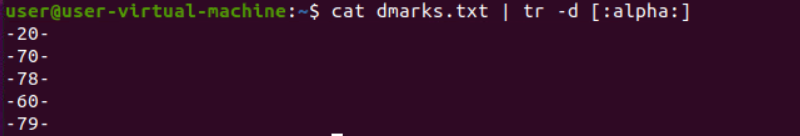
* cat filename.txt | tr -d [:digit:] : deletes the numbers from the given string.



* cat filename.txt | tr -d [:xdigit:] : eliminates the hexadecimal values.



* cat filename.txt | tr -d [:alpha:] : deletes all the characters from the given string.



* cat filename.txt | tr -cd [:alpha:] : removes all digits and parenthesis except characters.



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



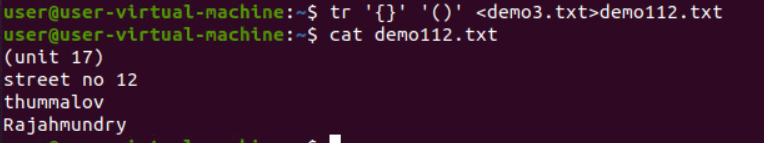
* $ echo “this is a string” | tr -s ‘ ’ ‘ ’ : To discard unwanted spaces



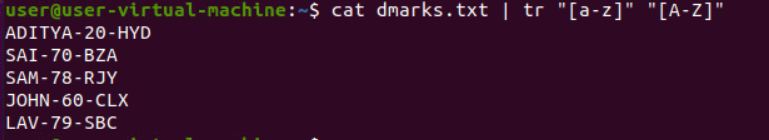
* tr -s ‘\n’ ‘ ’ < dmarks.txt : change new line character with space.



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



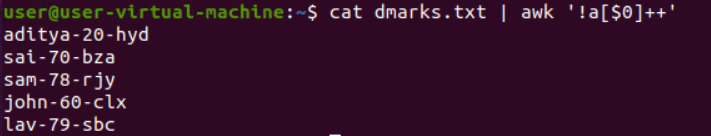
* cat filename.txt | tr “[a-z]” “[A-Z]” or cat filename.txt | tr “[:lower :]” “[:upper :]” : Converts lower case to upper case.



* cat filename.txt | tr [:space:] '\t' : translate white-spaces in a string to tabs.

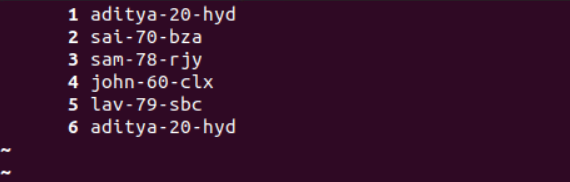


* cat dmarks.txt | awk ‘!a[$0]++’ : eliminate repeated charecters

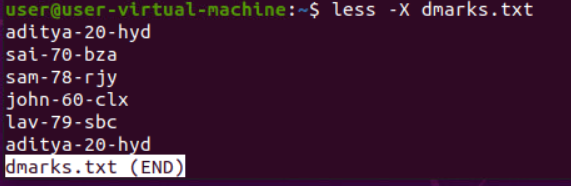


**LESS Command**

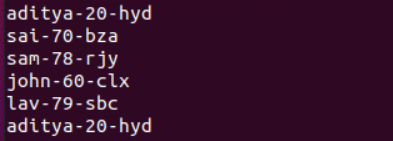
* less –N dmarks.txt : If you want to see the line numbers in the less command output



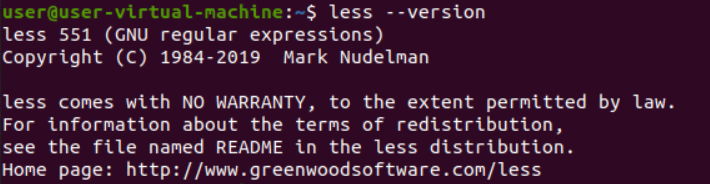
* less -X dmarks.txt : save the file name on screen when file exits.



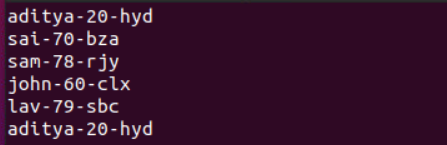
* less –S dmarks.txt : long line can be seen by side wrapping



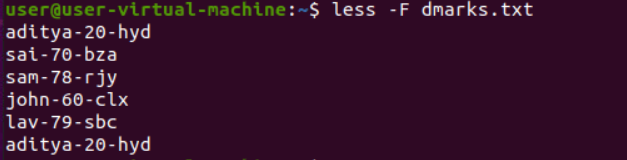
* less –version :



* less –m dmarks.txt : show more detailed prompt including file position.

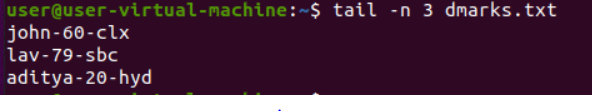


* less –F dmarks.txt



**TAIL Command**

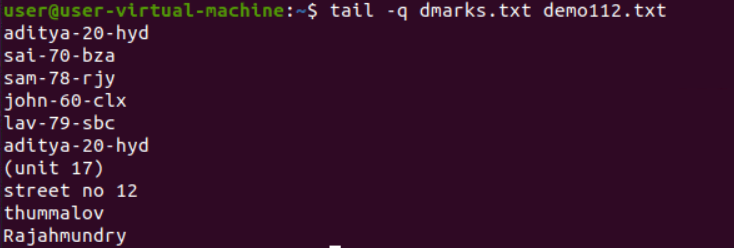
* tail –n 3 dmarks.txt : It prints the last num lines.



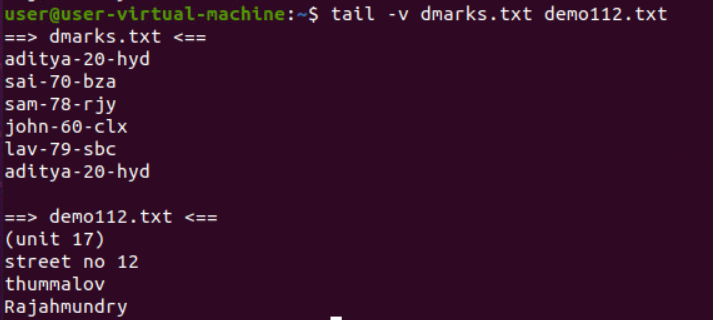
* tail –c 6 dmarks.txt : It prints last num bytes from the file specified.



* less –q dmarks.txt demo.txt : It is used if more than 1 file is given



* less –v dmarks.txt demo.txt : data from the specified file is always preceded by its file name.

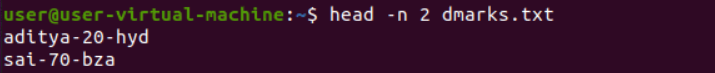


* tail –version :

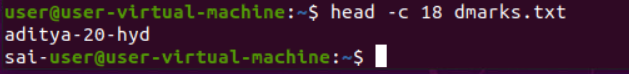


**HEAD Command**

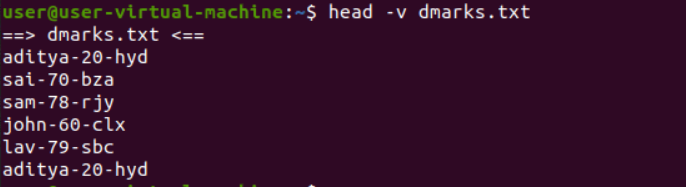
* head –n 2 dmarks.txt : print top 2 line in file.



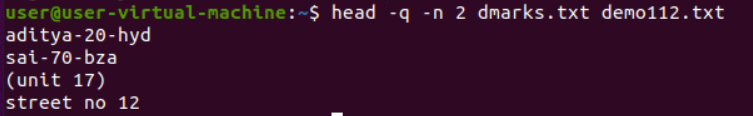
* head –c 18 dmarks.txt : print first 18 characters of the file.



* head –v dmarks.txt : display the file name along with contents.



* head –q –n 2 dmarks.txt demo112.txt : it is used for more than 1 file.



**FILE PERMISSIONS**

**CHMOD**

Symbol Permissions

--- no permission

--x execute

-w- write

-wx execute + write

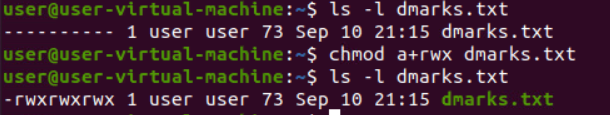
r-- read

r-x read + execute

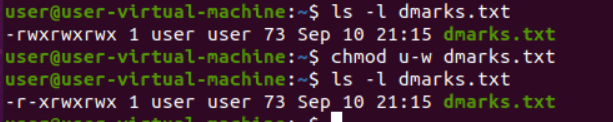
rw- read + write

rwx read + write + execute

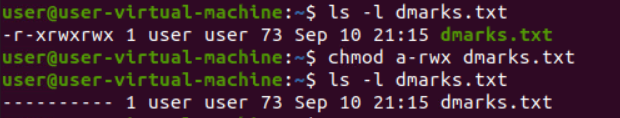
* chmod a+rwx dmarks.txt : it will make all the users to read, write and execute.



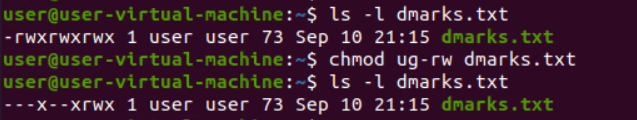
* chmod u-w dmarks.txt : it will remove write permission for user.



* chmod a-rwx dmarks.txt : it will remove all the read , write and execute permissions for all the users.



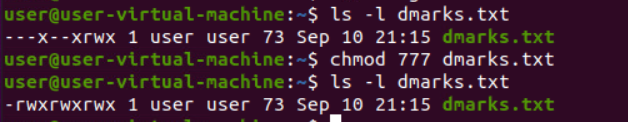
* chmod uq-rw dmarks.txt : it will remove the read and write permission for user and group.



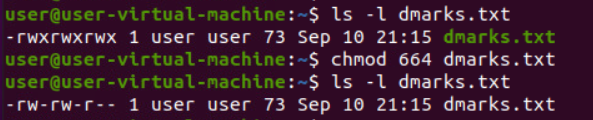
**Hexadecimal representation :**

1. no permission
2. execute
3. write
4. execute + write
5. read
6. read + execute
7. read + write
8. read + write + execute

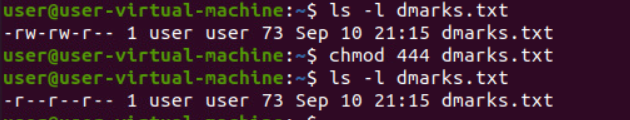
* chmod 777 dmarks.txt : it will give all the read, write and execute permissions to all users.



* chmod 664 dmarks.txt : it will give read and write permission to user and group and read permission to others.



* chmod 444 dmarks.txt : it will give read permission to user, group and others.



* chmod 000 dmarks.txt : it remove all the permission to all the users.

