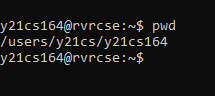
**SKILL ORIENTED COURSE**

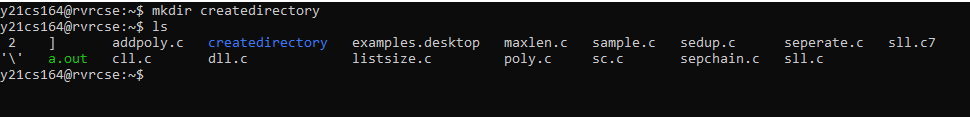
**CSSL2 - LINUX PROGRAMMING**

**MODULE-1**

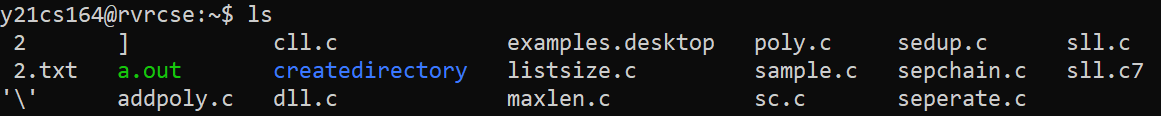
**DIRECTORY RELATED**

**UTILITIES**

1. ***pwd* :** It is used to print the full path of the present working directory starting from the root.****
2. ***mkdir* :** It is used to create directory(ies), if they do not exist.



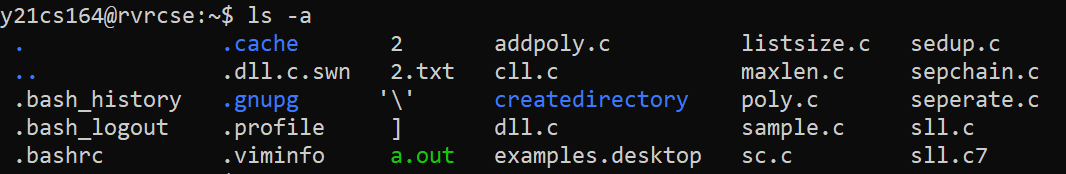
1. ***ls* :** It is used to list information about the files of the current directory by default (or) a specified directory.

**Syntax:** ls [Options] [File] 

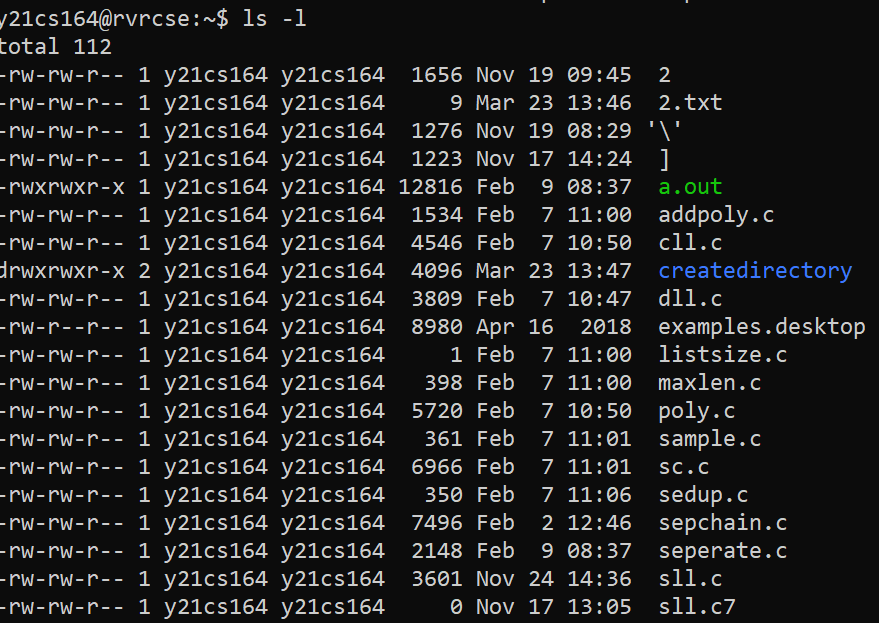
Some practical examples of ls command are shown below:

|  |  |
| --- | --- |
|  |  |

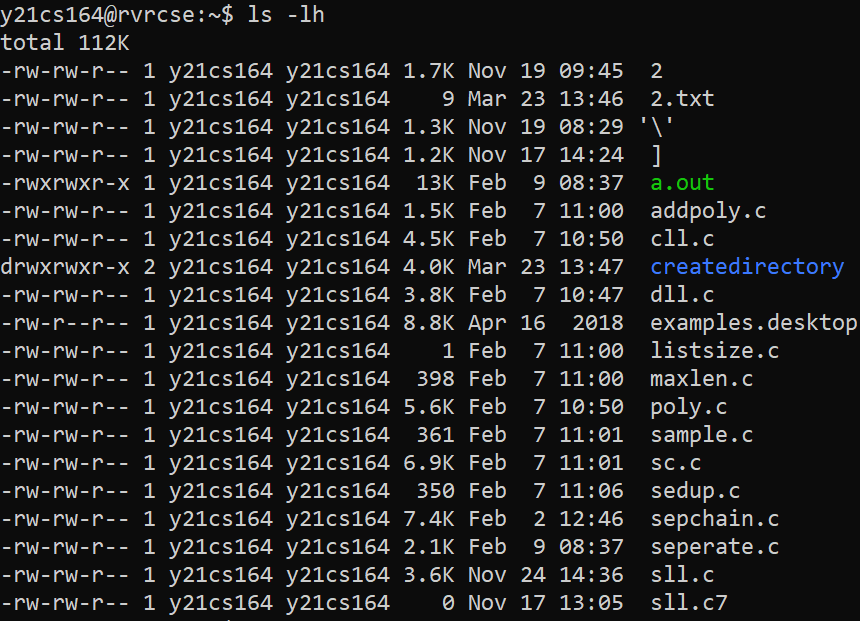
* ls –a : list all files including hidden file starting with '.'.



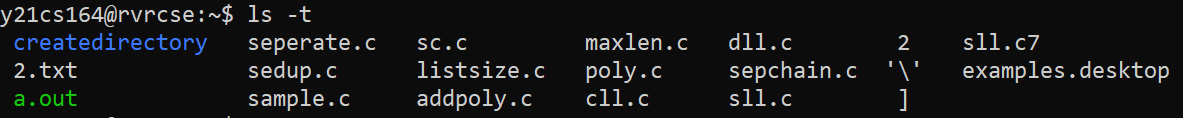
* ls –l : list with long format - show permissions.



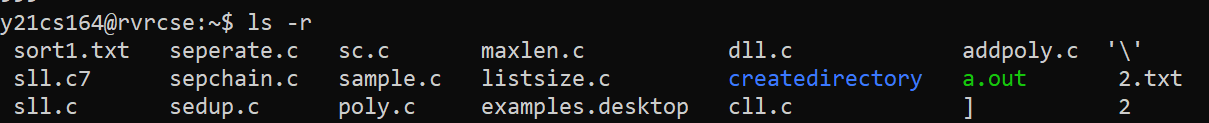
* ls –lh : This command will show you the file sizes in human readable format.



* ls –t : sort by time & date.

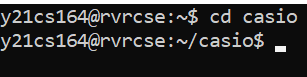


* ls –r : list in reverse order.



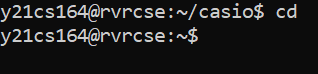
1. ***cd*:**It is used to change the directory to the home directory by default (or) to a specified directory.

* **To move inside a subdirectory:  
  Syntax: $ cd [directory\_name]**



* To change current working directory:

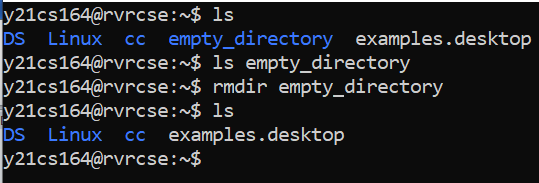
Syntax: $ cd



1. ***rmdir*:**It is used to remove directory(ies) only if they are empty.

* **Syntax** : rmdir [directory name]
  + If the directory still contains files or subdirectories, the rmdir command does not remove the directory.
  + To remove a directory and all its contents, including any subdirectories and files, use the rm command with the recursive option, **-r**.

**Syntax** : rm –r [directory name]



**MODULE-2**

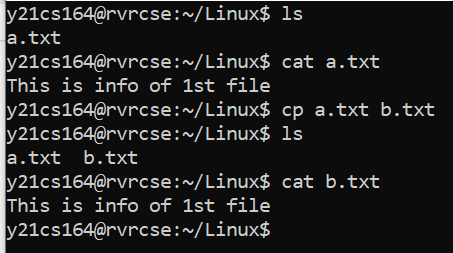
**FILE HANDLING AND**

**TEXT PROCESSING**

1. ***cp* :** It is sued to copy files or directories from a source to a destination in the file system.

* The cp command copies the source file specified by the SourceFile parameter to the destination file specified by the TargetFile parameter.
* If the target file exists, cp overwrites the contents, but the mode, owner, and group associated with it are not changed.

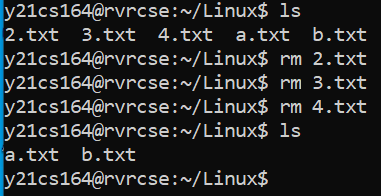
**Syntax :** cp [old file] [new file]

****

1. ***rm*** :

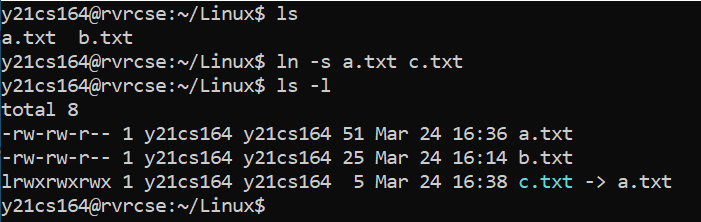
The rm command removes the entries for a specified file, group of files, or certain select files from a list within a directory.

**Syntax:** rm [filename]

****

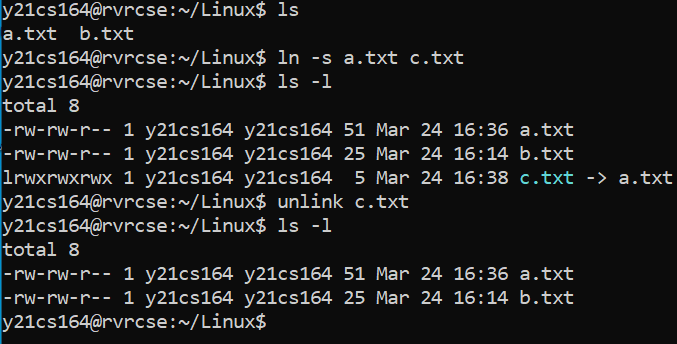
1. ***ln* :**It is used to create symbolic links between two files

**Syntax :**ln -s [old file] [new file]

****

1. ***unlink* :**It is used to remove the symbolic links between files

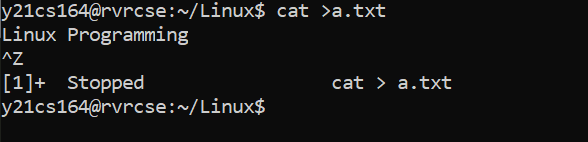
**Syntax :**unlink [filename]



1. **cat :** It is used to concatenate files and print on the standard output.

**Syntax:**

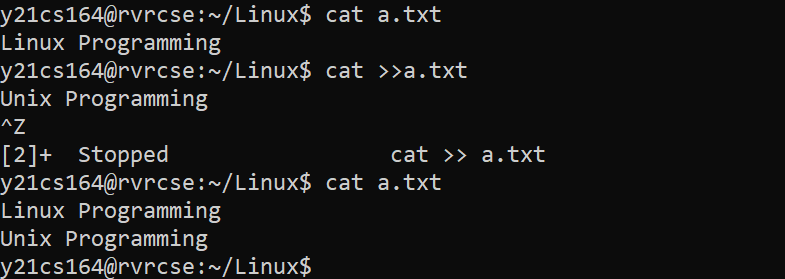
* for creating a new file and write content to it : - cat > [filename]

****

* Prints the contents of a file on standard output :- cat [filename]



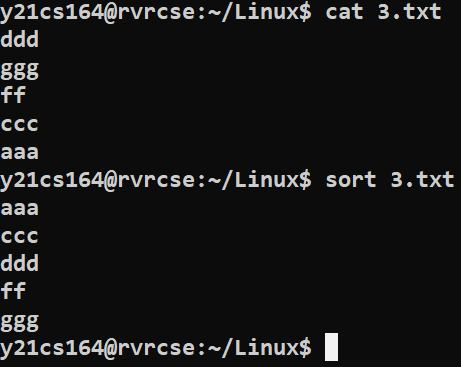
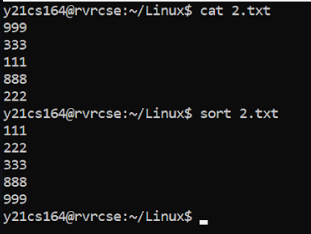
* Used to append extra content to theter file :- cat >> [filename]

****

1. ***sort*  :**

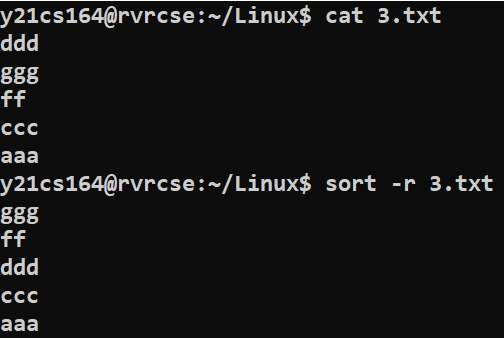
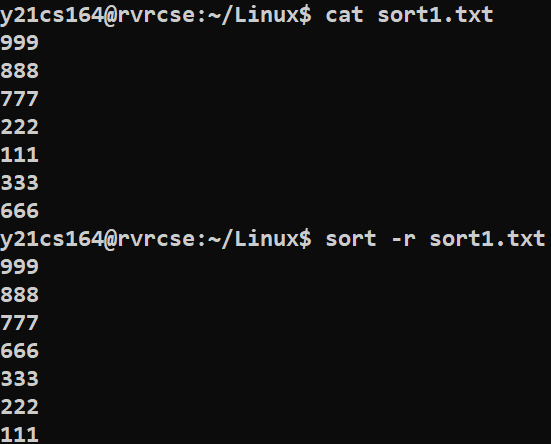
* The sort command is a tool for sorting file contents and printing the result in standard output. Reordering a file's contents numerically or alphabetically and arranging information in ascending or descending order improves readability.
* For sorting of strings and numbers in ascending order:

**Syntax:** sort [filename]



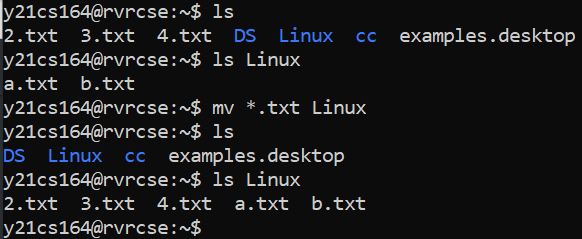
* For sorting of strings and numbers in descending order:

**Syntax:** sort –r [filename]



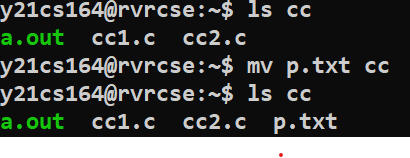
1. ***mv* :**mv stands for move and it is used as follows:
   * To move one or more files from one directory to another directory.

**Syntax:** mv \*.[filetype] [Destination]

****

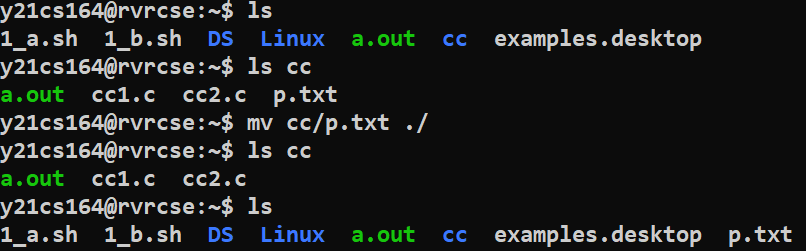
* To move a file into a directory:

**Syntax:** mv [file\_name] [directory\_name]



* To move a file outside a directory:

**Syntax:** mv [directory\_name]**/**[file\_name] **./**

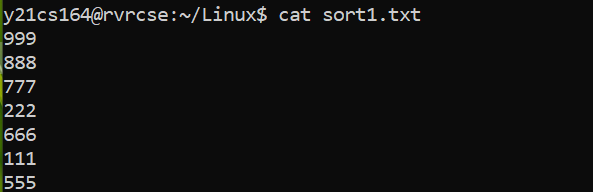


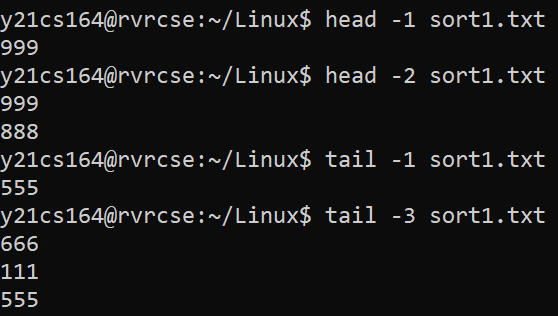
1. ***head & tail*:**

* As their names imply, the head command will output the first part of the file, while the tail command will print the last part of the file. Both commands write the result to standard output.

**Syntax :** Head:  head  [number of lines]   [filename]

    Tail:  tail  [no of lines]  [filename]

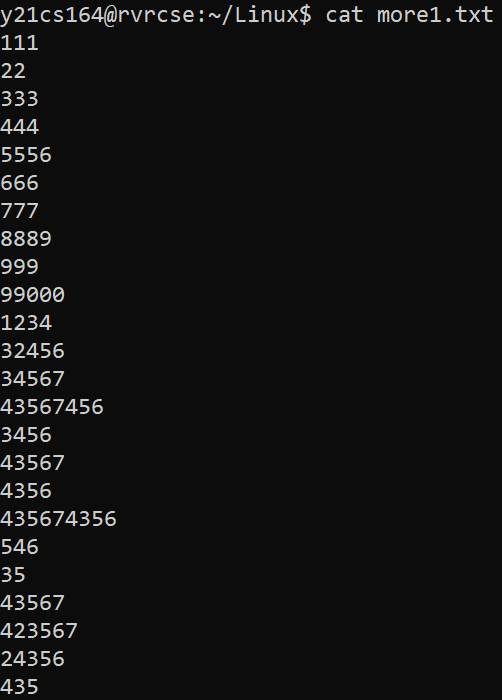
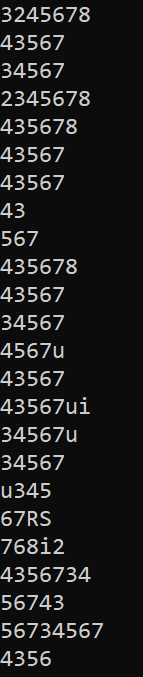
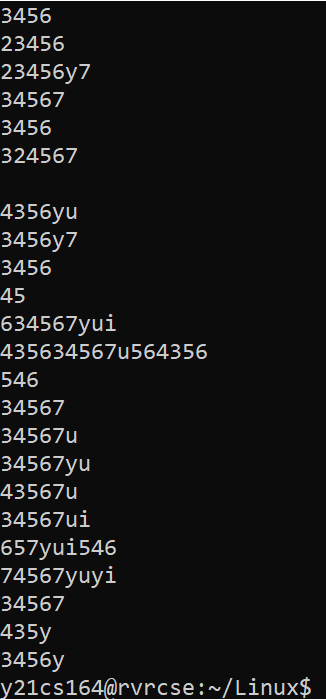


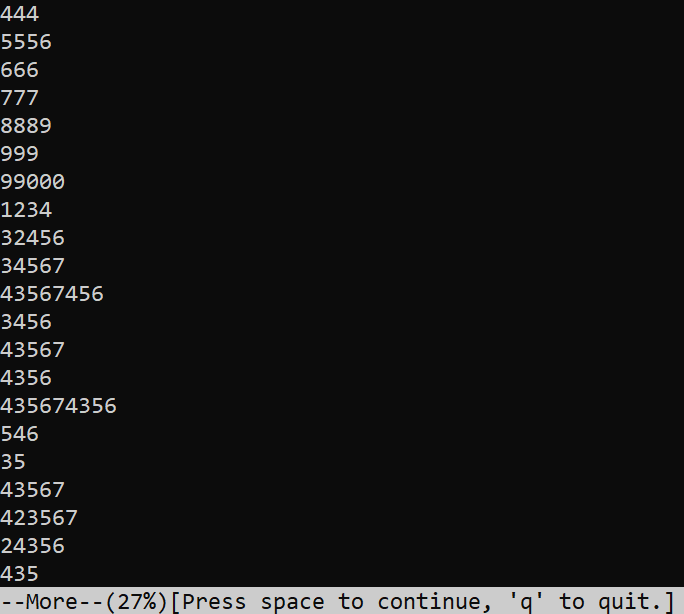
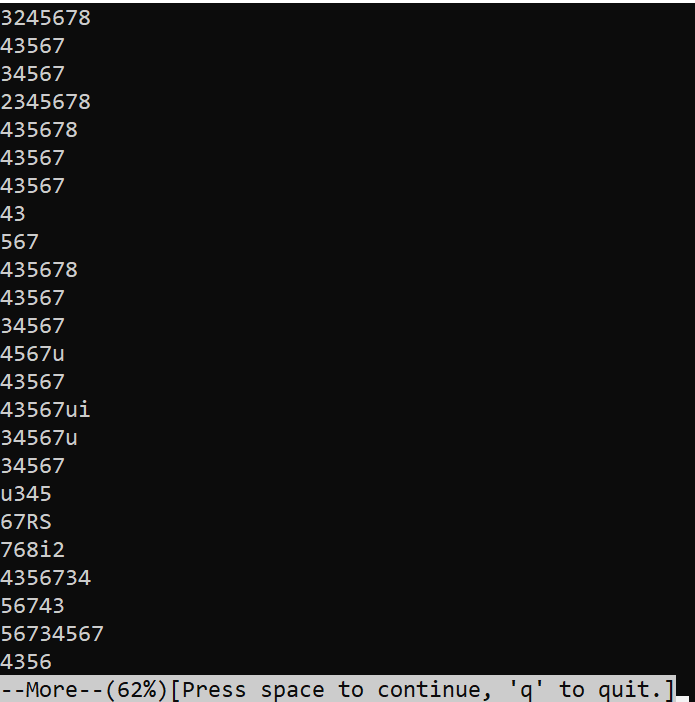
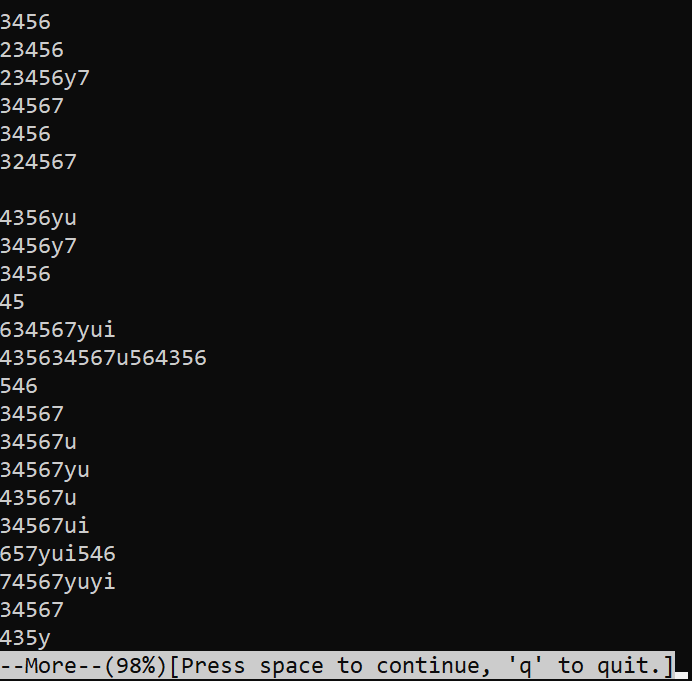


1. ***more* :**

The more command reads files and displays the text one screen at a time. The command pauses after each screen and prints the word More at the bottom of the screen.

**Syntax :** more   -d   [filename]

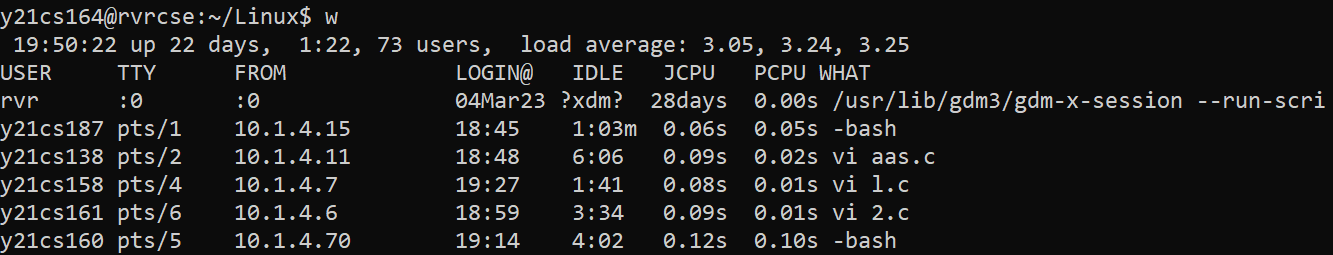
**  **

**  **

1. ***w* :**

The "w" command displays information about all users logged into the system currently, and their processes..

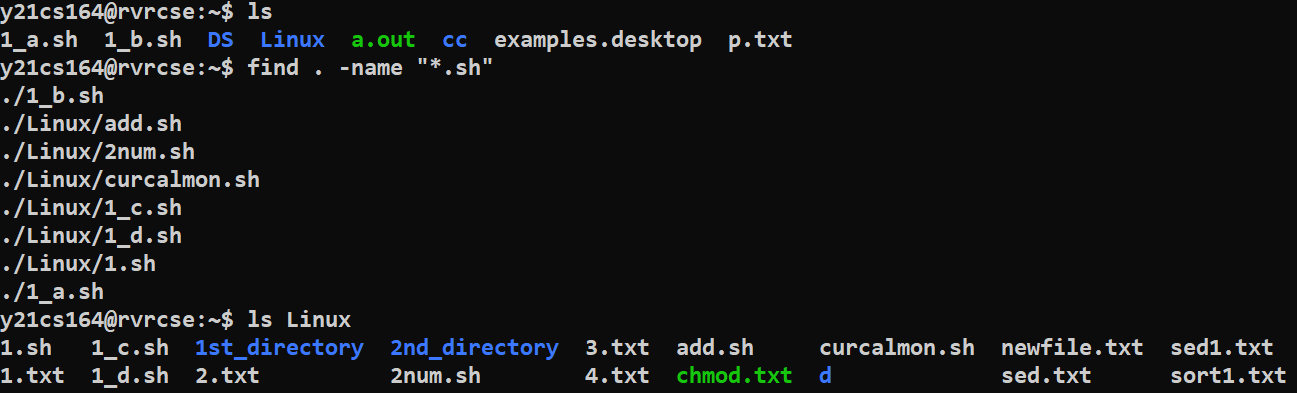
**Syntax :**w

****

1. ***find* :** It is used to find files or directories in a specific directories by,

* Specific file type:

**Syntax :** find **.** –name “\***.**[file\_type]”



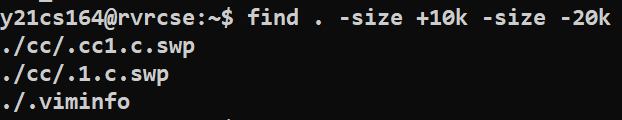
* Specific file size

**Syntax:** find **.** –size [sign][number][byte\_type]

Here , [sign] 🡪 ‘+’ (used for files greater than) or ‘-‘(used for files lesser than)

[number] 🡪specify the size

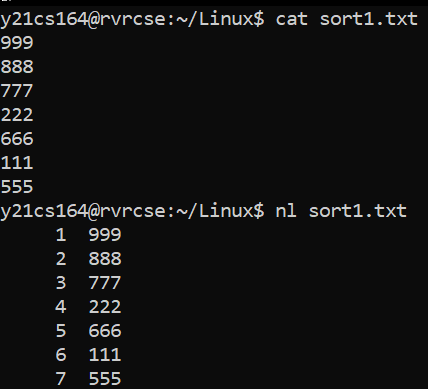
[byte\_type] 🡪 ‘k’ for kilobytes, ‘M’ for Megabytes, and ‘G’ for Gigabytes



1. ***nl*:**

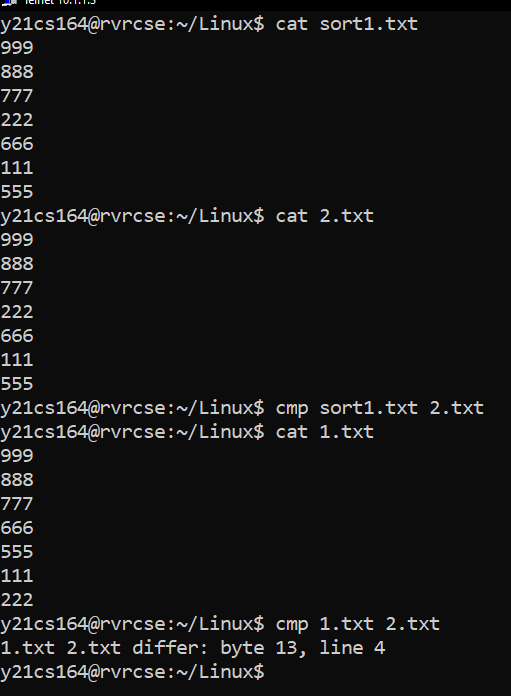
The nl command reads the File parameter (standard input by default), numbers the lines in the input, and writes the numbered lines to standard output.

**Syntax :** nl [filename]

****

1. ***cmp*:** It compares two files byte by byte and returns at which byte the files first differ standard.

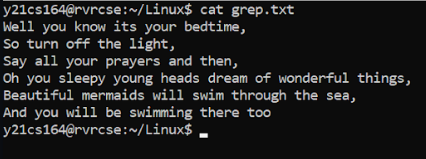
**Syntax:** cmp [file1] [file2]



1. ***grep*:**  It searches for a pattern in the given files:

* *–wn* options are used to restricts matching to whole words only with line numbers

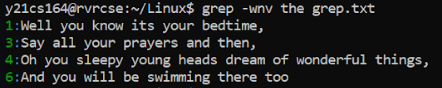
**Syntax :** grep –wn [word] [filename]





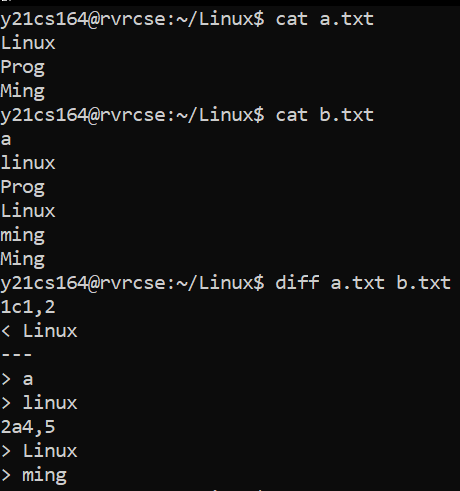
* -wnv option is used to display only those lines in a file don’t match,

**Syntax:** grep –wnv [word] [filename]

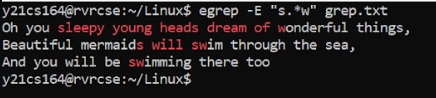


1. ***diff*:** The diff command analyzes two files and prints the lines that are different. It displays three kinds of editing changes: ‘a’ adding lines, ‘c’ changing lines and ‘d’ deleting lines

**Syntax :** diff [file1] [file2]

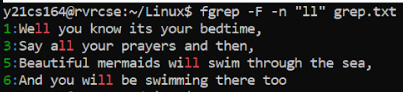
****

1. ***egrep*:** It is used to search for extended regular expression patterns:

****

****

1. ***fgrep*:** It is used to search for fixed strings in the specified file.



1. ***Chmod*:**

chmod command is used to change the access permissions of files and directories. It stands for change mode. It can not change the permission of symbolic links.

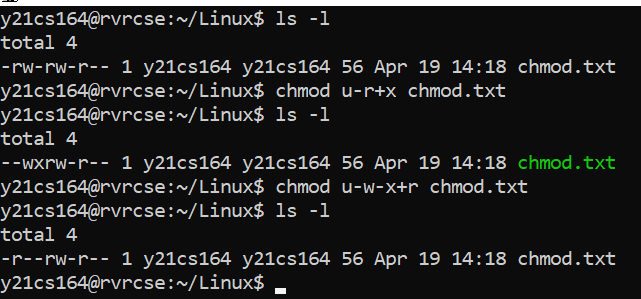
**Syntax :**

* To Remove **:**chmod [to whom]-[command] [filename]
* To add : chmod [to whom]+[command] [filename]

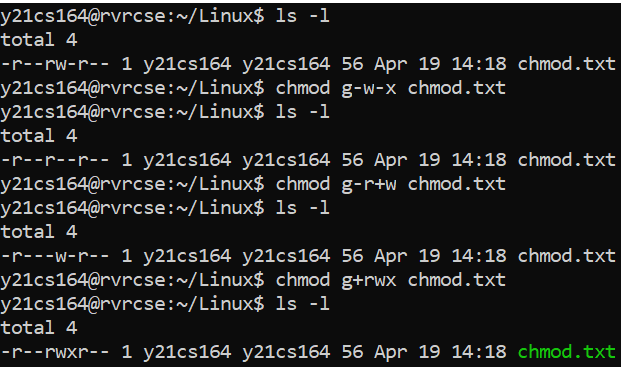
To whom : u – user ,g – group ,o-owner.

Command : r – read ,w – write ,x – execute.

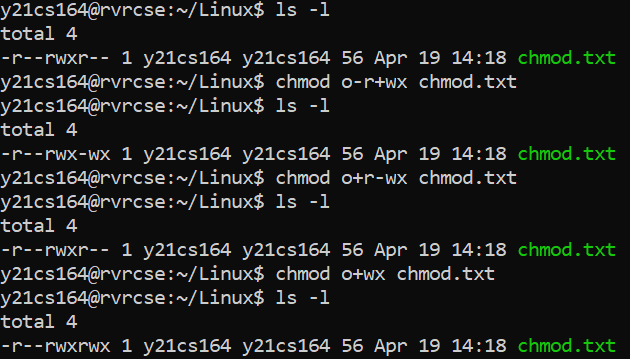
* user:



* group:



* owner:



1. ***chgrp*:**

chgrp command is used to change the group ownership of a file or directory.

**Syntax :** chgrp [groupname] [filename]

1. ***chown*:**

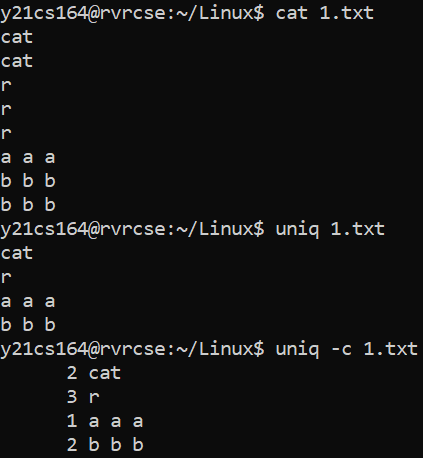
The chown command changes the owner of the file or directory specified by the File or Directory parameter to the user specified by the Owner parameter.

**Syntax :** chown [username] [File name]

1. ***uniq*:**

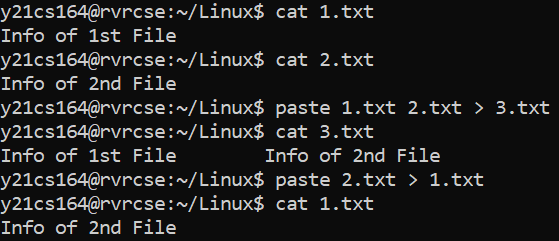
The uniq utility displays a file with all of its identical adjacent lines replaced by a single occurrence of the repeated line.

**Syntax :**uniq [option] [filename]



1. ***paste*:** It is used to parallel merge or join two files by outputing lines consisting of each line separated by **tab** delimiter

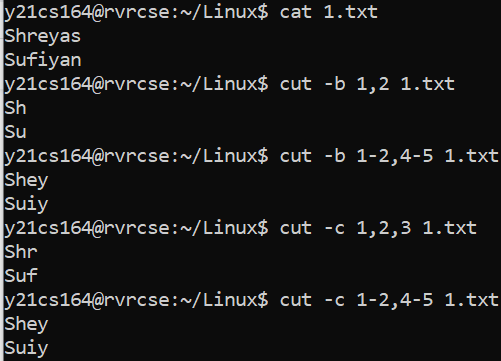
**Syntax :** paste [filename1] [filename2] > [newfile]



1. ***cut*:** It is used for cutting out the sections from each line and displaying on standard output.

**Syntax :** cut [option] [data sections] [filename]

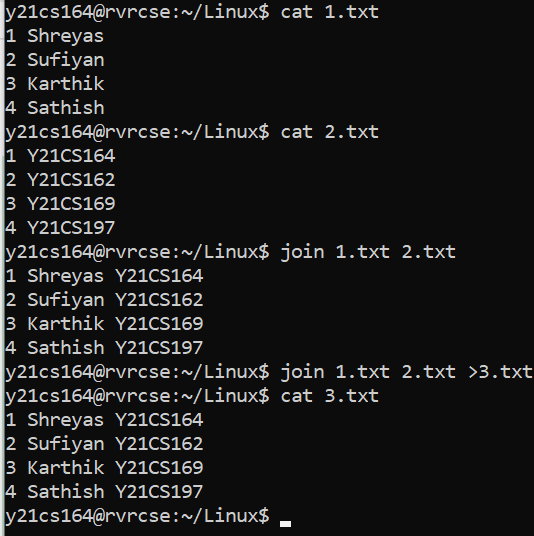
Here, [option] 🡪 ‘-b’(to cut specific bytes of each line in a file) or ‘-c’(to specify the columns to print in each line of a file.



1. ***join*:**

The join command reads the files specified by the File1 and File2 parameters, joins lines in the files according to the flags, and writes the results to standard output. The File1 and File2 parameters must be text files.

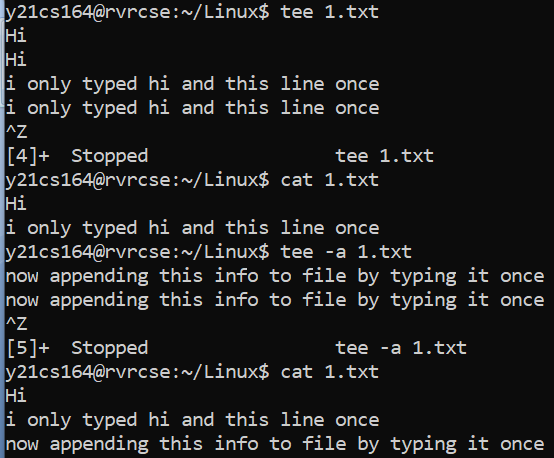
**Syntax :** join [file1] [file2]



1. ***tee*:** It is used to read from standard input and write to standard output and files

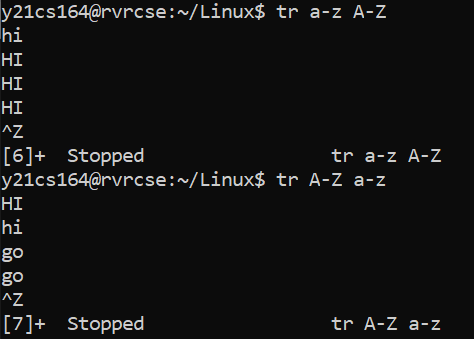
**Syntax :** tee [option] [filename]

Here, [option] 🡪 without any options or ‘-a’(used to append the standard input to a file rather than overwriting)



1. ***tr*:**  It is used to perform text trasformations, i.e it transforms set1 type text into set2 type texts after typed.

**Syntax :**tr [set1] [set2]

****

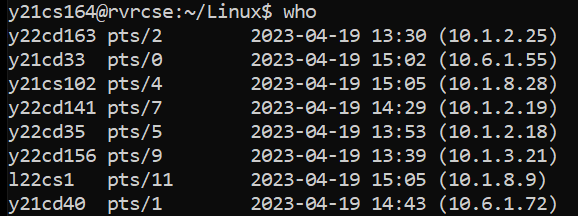
**MODULE-3**

**DISK UTILITIES, BACKUP AND OTHER UTILITIES**

1. ***who*:**

who command is a tool print information about users who are currently logged in.

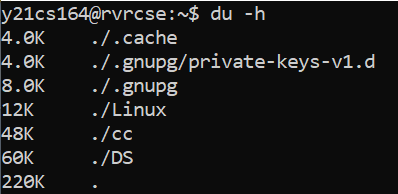
**Syntax :** who



1. ***du*:**

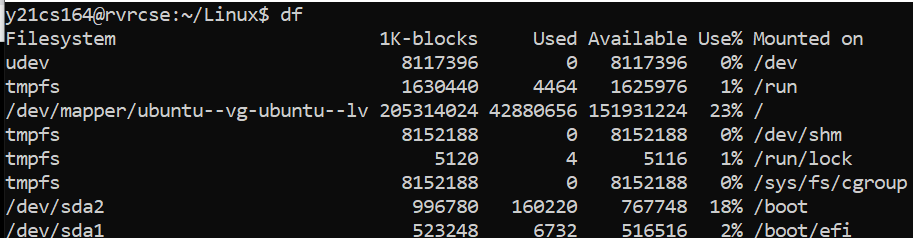
It is used to summarize disk usage of the set of files, recursively for directories.

**Syntax:**du –h



1. ***df*:** It is used to summarize disk usage of the set of files, recursively for directories.

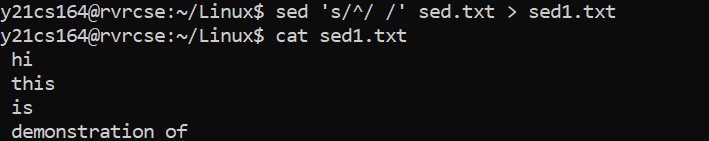
**Synatx**: df



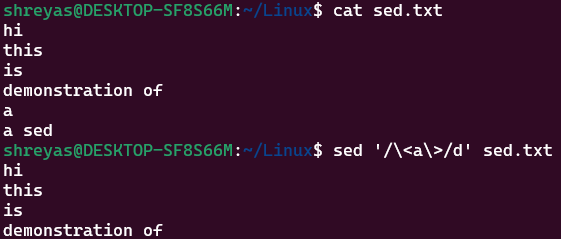
1. ***sed*:**

The stream editor utility sed scans one or more files and performs an editing action on all ofthe lines that match a particular condition**.**

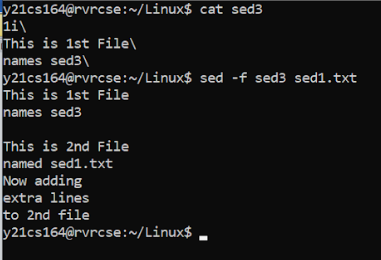
* Substituting text: Substituting first character with a space in each line



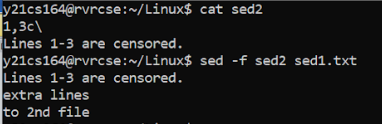
* Deleting text: Deleting only those lines that contain the word ‘a’



* Inserting text:



* Replacing text:

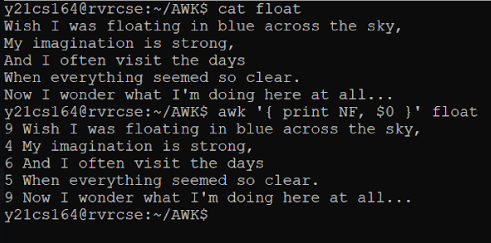


**MODULE-4**

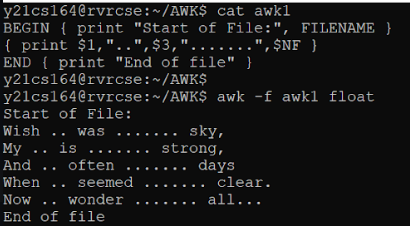
**PROGRAMMABLE TEXT PROCESSING**

***awk*** is a programmable text-processing utility that scans the lines of its input and performs actions on every line that matches a particular criterion.

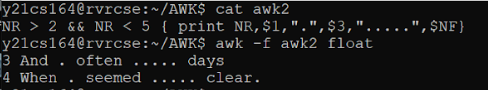
1. **Accessing individual files:**

****

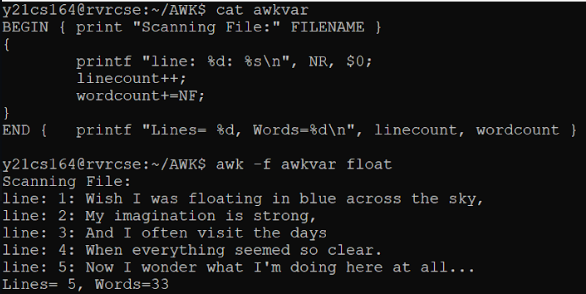
1. **Begin and End:**

****

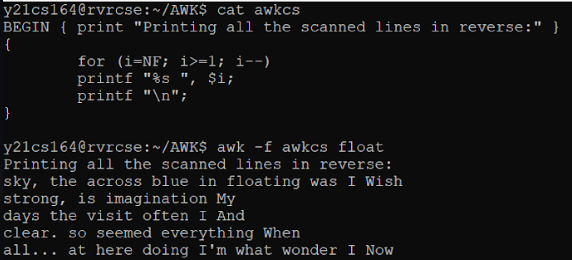
1. **Operators:**

****

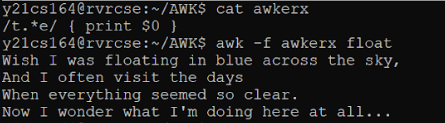
1. **Variables:**

****

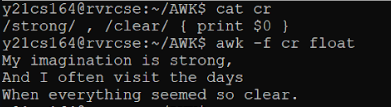
1. **Control Structures:**

****

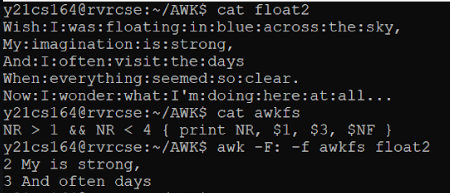
1. **Extended Regular Expressions:**

****

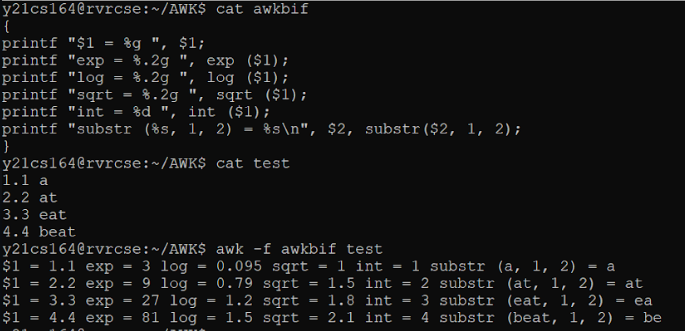
1. **Conditional Ranges:**

****

1. **Field Separators:**

****

1. **Built-In Functions:**

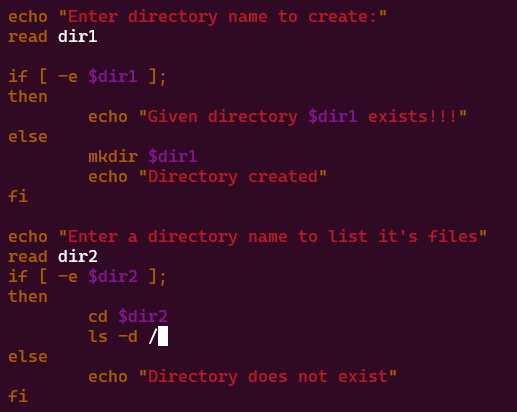
****

**MODULE - 4**

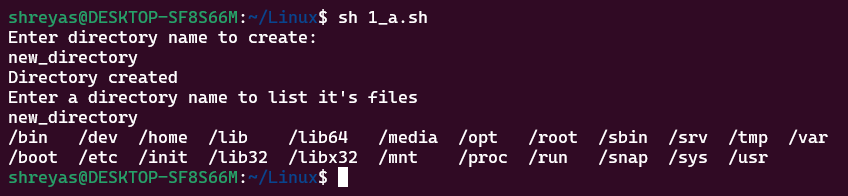
**SHELL SCRIPTING**

1. Write a shell script program for the following:
2. To create a directory and list all the directory files in a directory:

Code:

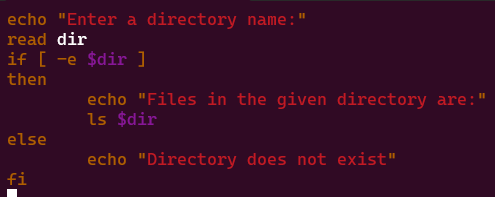


Output:

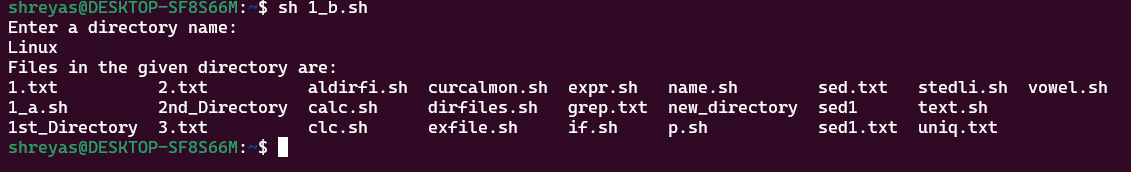


1. To display a list of all the files in the current directory.

Code:

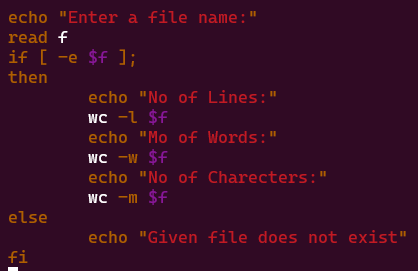


Output:

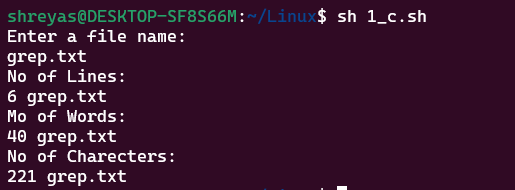


1. To count no of lines, words, and characters of an input file.

Code:

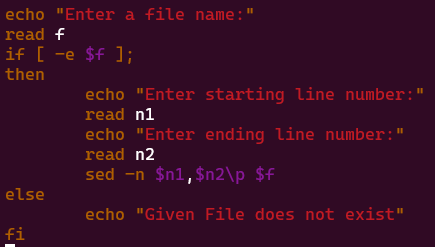


Output:

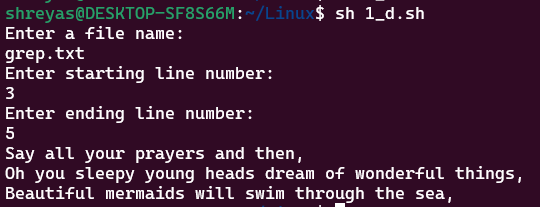


1. To accept a file name starting and ending line numbers as arguments and display all the lines between given line numbers.

Code:

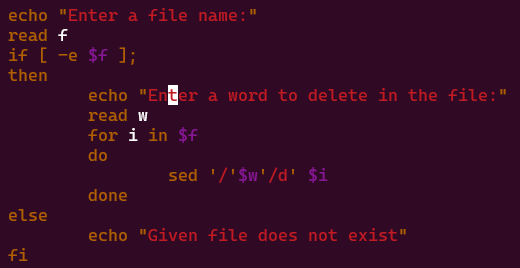


Output:

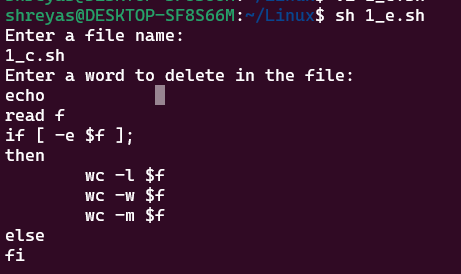


1. To deletes all lines containing the specified word in one or more files supplied as arguments to it.

Code:

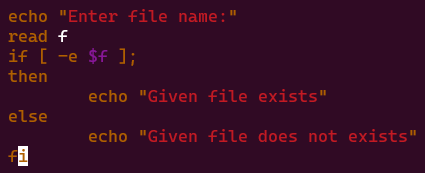


Output:

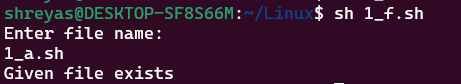


1. To test whether the given file is existing or not.

Code:

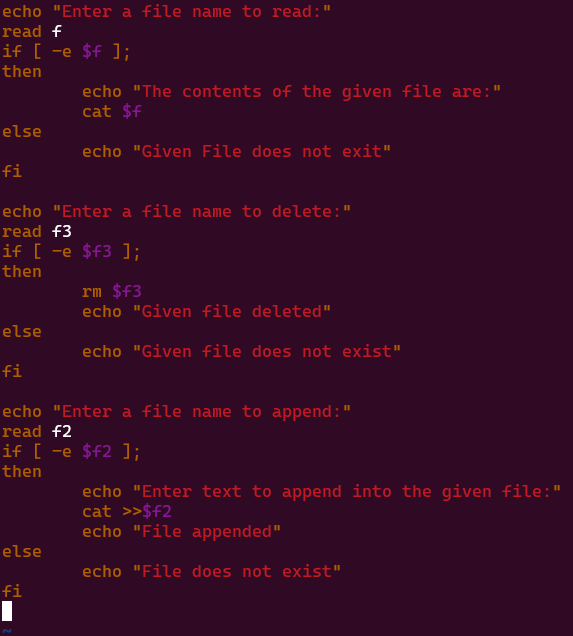


Output:

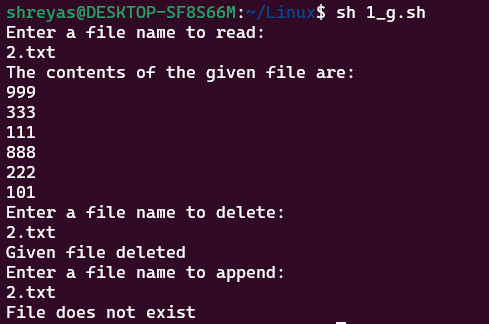


1. To read, delete and append a file.

Code:

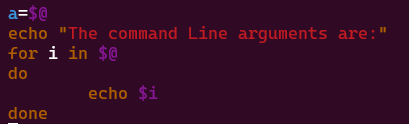


Output:

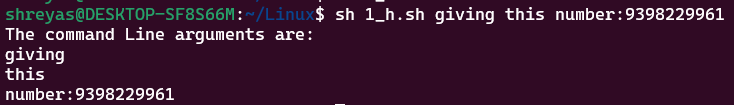


1. To store all command line arguments to an array and print.

Code:

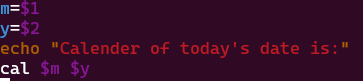


Output:



1. To print the calendar month by default.

Code:



Output:

