**SKILL ORIENTED COURSE**

**CSSL2 – LINUX PROGRAMMING**

**MODULE-1**

**DIRECTORY RELATED**

**UTILITIES**

1. **pwd :**

The **pwd(**print working directory) command writes to standard output the full path name of your current directory (from the root directory).

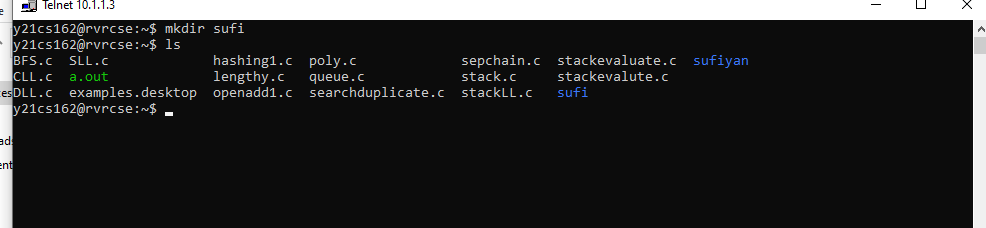
**Syntax- pwd** [ [**-L**](https://www.ibm.com/docs/en/aix/7.1?topic=p-pwd-command#pwd__pwd_flags_bigl) | [**-P**](https://www.ibm.com/docs/en/aix/7.1?topic=p-pwd-command#pwd__pwd_flags_bigp) ]

Screenshot (3).png

1. **mkdir :**

**mkdir** command in Linux allows the user to create directories (also referred to as folders in some operating systems ).

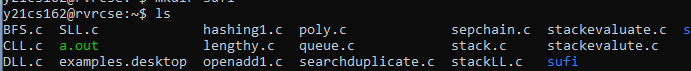
**Syntax -** mkdir [options...] [directories ...]



1. **ls :**

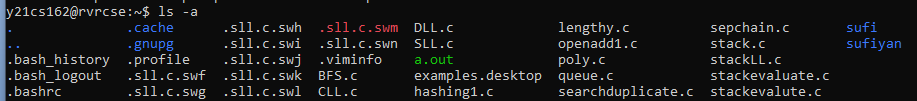
It's used to display a list of files and sub-directories in the current directory

**Syntax -** ls[ Options ] [File]

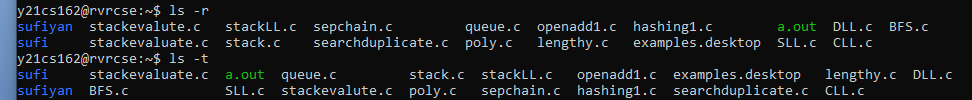


There are few commands in ls :

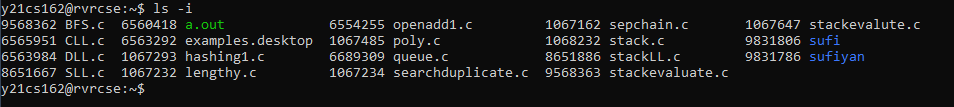
1. ls - a : list all files including hidden file starting with '.'.



1. ls - d : list directories - with ' \*/'.
   1. Screenshot (6)2.png
2. ls- r : list in reverse order.
3. ls- t : sort by time & date.



1. ls- i : list files inode(index) number.



1. **cd :**

The cd command in Linux stands for change directory. It is used to change the current directory of the terminal.

**Syntax - $ cd [directory]**

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

Screenshot (7).png

To return to main directory : $ cd

Screenshot (4).png

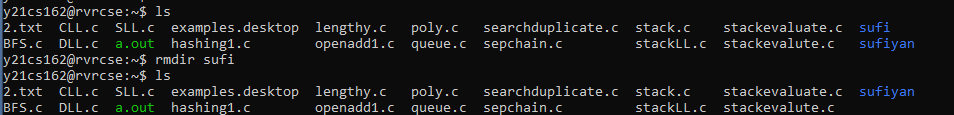
1. **rmdir :**

* The rmdir command removes the directory, specified by the Directory parameter, from the system.

**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]



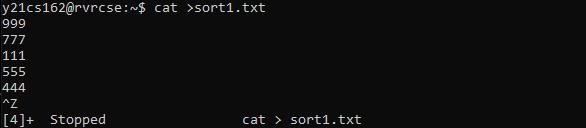
**Module – 2**

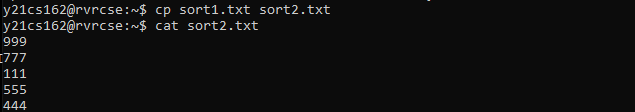
**File Handling and Text Processing**

1. **cp :**

* 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]

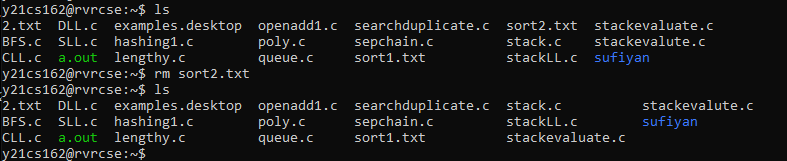




1. **rm** :

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

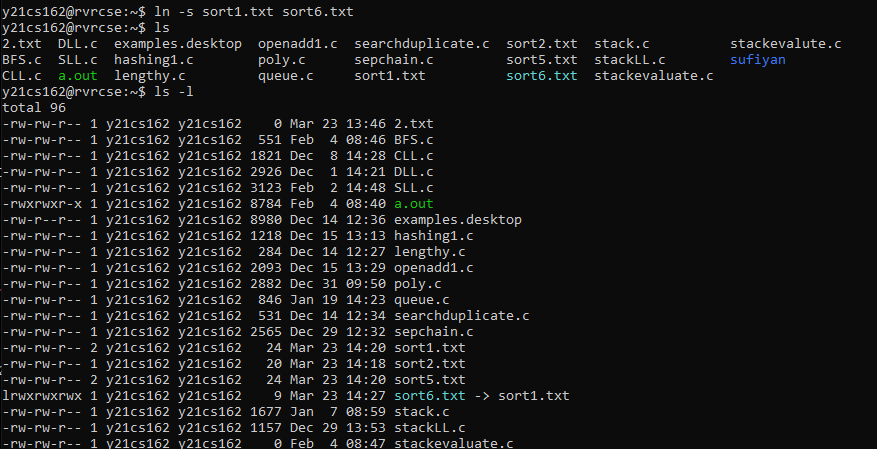
**Syntax:** rm [filename]



1. **ln :**

The ln command links the file designated in the SourceFile parameter to the file designated by the TargetFile parameter or to the same file name in another directory specified by the TargetDirectory parameter. By default, the ln command creates hard links.

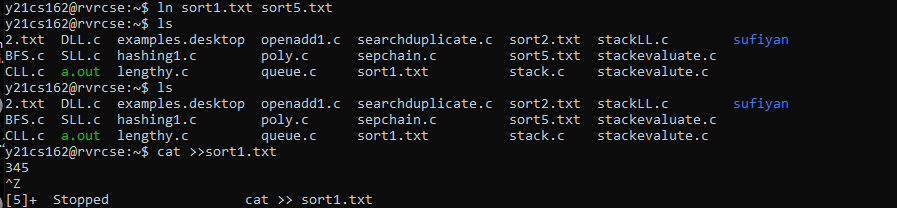
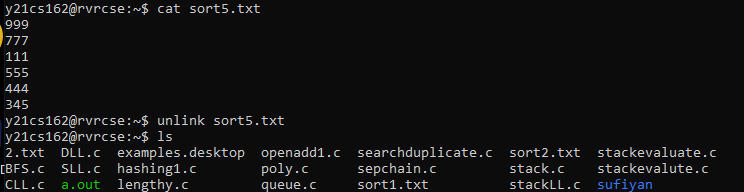
**Syntax :**ln-s [old file] [new file]



1. **unlink :**

The unlink command performs the unlink subroutine on a specified file.

**Syntax :**unlink [filename]

****

1. **cat :**

The cat command reads each File parameter in sequence and writes it to standard output. If you do not specify a file name, the cat command reads from standard input.

**Syntax:**

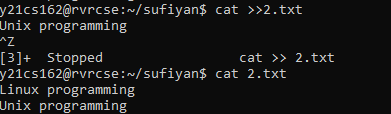
* for creating a file : - cat > [filename]

****

* for displaying a file :- cat [filename]



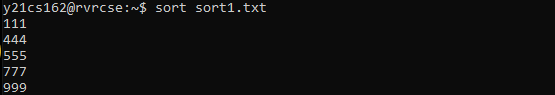
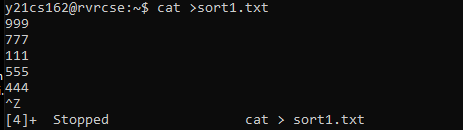
* for appending data in file :- cat >> [filename]



**6 .sort :**

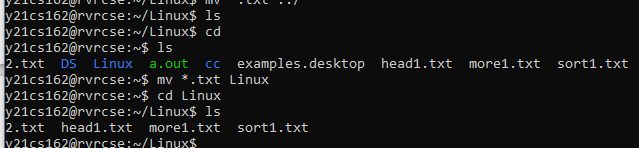
The sort command is a tool for sorting file contents and printing the result in standard output. Reordering a file's contents numerically or alphabetically and arranging information in ascending or descending order improves readability.

**Syntax :**sort [filename]



1. **. mv :**
   * **mv** stands for **move**. mv is used to move one or more files or directories from one place to another in a file system.
   * It moves a group of files to a different directory.

**Syntax** : mv [source] [ Destination]

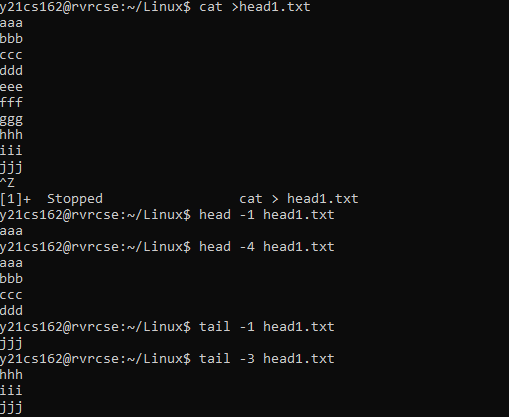
****

**8. 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]

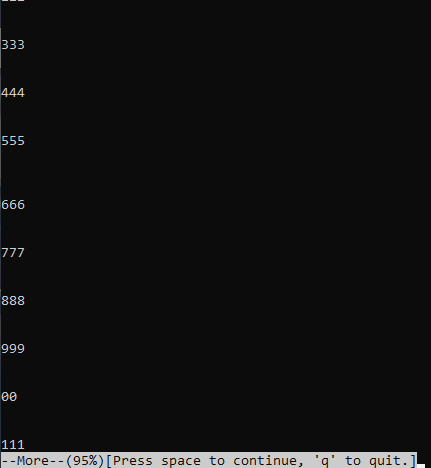
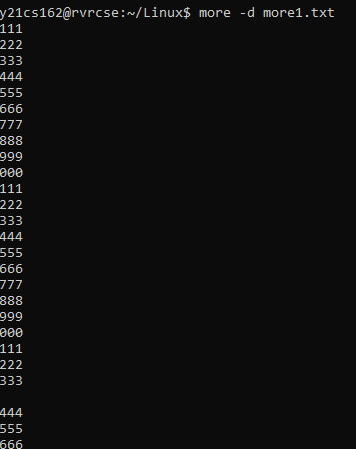


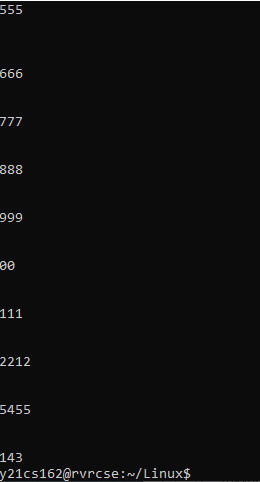
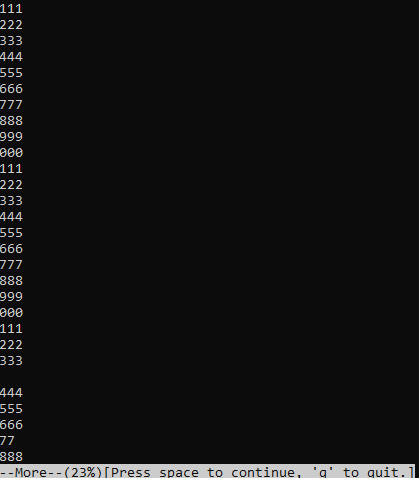
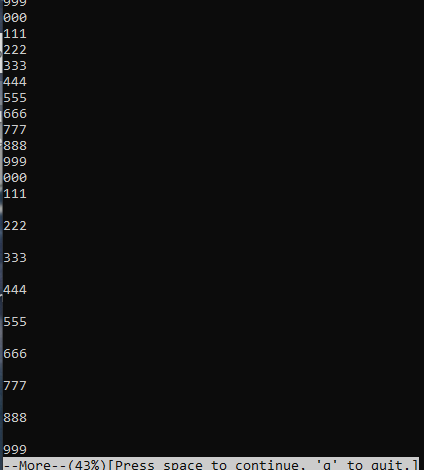
**9.more :**

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]

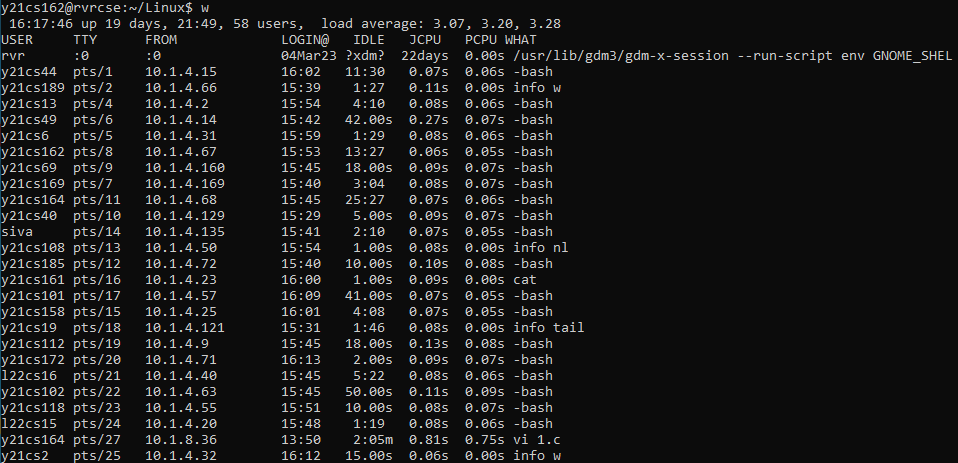




**10 .w :**

The "w" command displays information about all users logged into the current system.

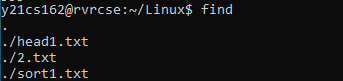
**Syntax :**w



1. **find :**

* The find command in UNIX is **a command line utility for walking a file hierarchy**.
* It can be used to find files and directories and perform subsequent operations on them.
* It supports searching by file, folder, name, creation date, modification date, owner and permissions.

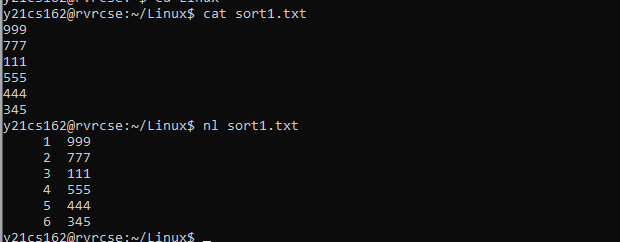
**Syntax :**find



1. **nl :**

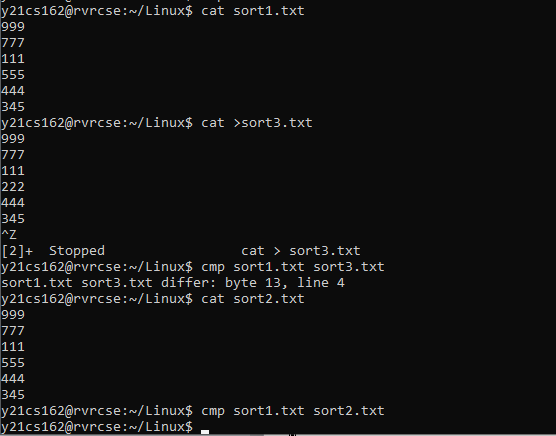
The nl command reads the File parameter (standard input by default), numbers the lines in the input, and writes the numbered lines to standard output.

**Syntax :**nl [filename]

****

1. **cmp:**

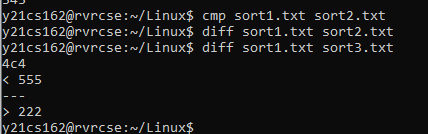
The cmp command compares files designated by the File1 and File2 parameters and writes the results to standard output.

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

1. **diff :**

the diff command analyzes two files and prints the lines that are different.

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

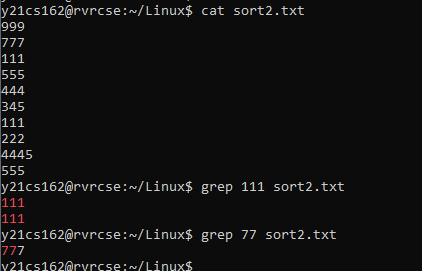
****

**15.grep:**

Grep is a useful command to search for matching patterns in a file. grep is short for "global regular expression print".

Data should be simple to work on it.

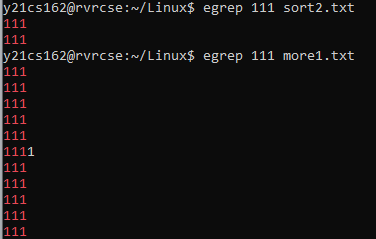
**Syntax :**grep [what to find] [filename]

****

1. **egrep:**

* **egrep** is a pattern searching command which belongs to the family of [grep](https://www.geeksforgeeks.org/grep-command-in-unixlinux/) functions.
* It works the same way as ***grep -E*** does. It treats the pattern as an extended regular expression and prints out the lines that match the pattern.
* If there are several files with the matching pattern, it also displays the file names for each line.
* Can work on complicated data

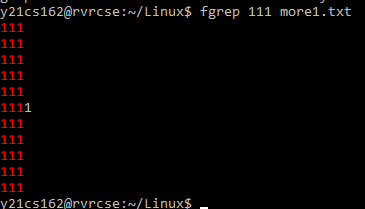
**Syntax:**egrep [PATTERN] [files]

****

1. **fgrep :**

The fgrep command searches specifically for **Pattern** parameters that are fixed strings. The fgrep command displays the file that contains the matched line if you specify more than one file in the File parameter.

**Syntax :**fgrep [pattern] [filename]

****

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 :**

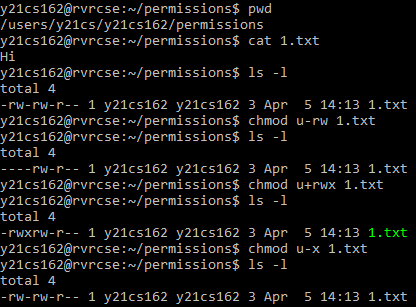
* To Remove **:**chmod [to whom]-[command] [filename]
* To add : chmod [to whom]+[command] [filename]

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

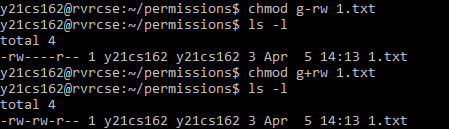
Command : r – read ,w – write ,x – execute .

:

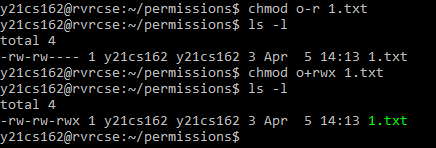
1.user :



2. group :



3. owner :



**19 .chgrp :**

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

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

**20 .chown :**

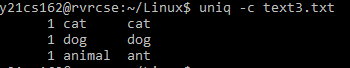
The chown command changes the owner of the file or directory specified by the File or Directory parameter to the user specified by the Owner parameter.

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

**21.uniq:**

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

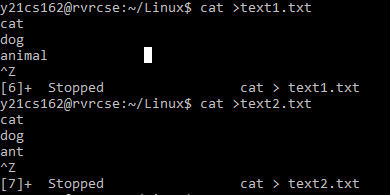
**Syntax :**uniq [option] [filename]

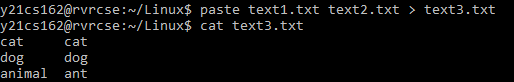


**22. paste :**

It is used to join files horizontally (parallel merging) by outputting lines consisting of lines from each file specified, separated by **tab** as delimiter, to the standard output

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



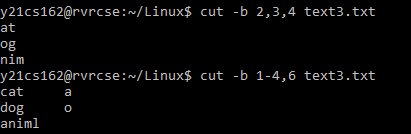


**23. cut:**

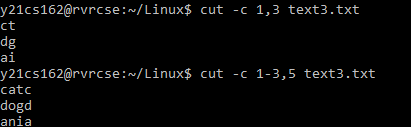
The cut command is a command-line utility that allows you to cut out sections of a specified file or piped data and print the result to standard output.

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

1. By bytes :



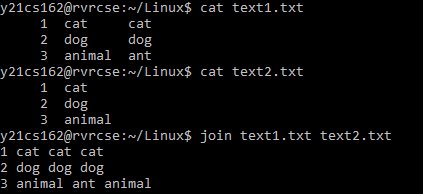
1. By columns:



**24** .**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.

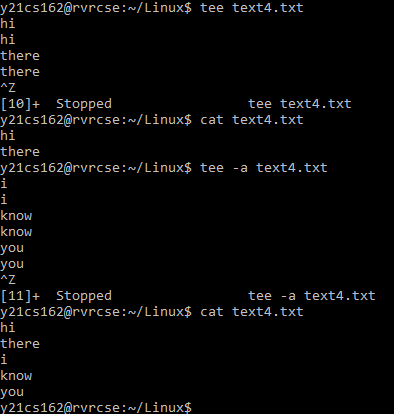
**Syntax :**join [file1] [file2]



**25 .tee :**

The tee command, used with a pipe, reads standard input, then writes the output of a program to standard output and simultaneously copies it into the specified file or files.

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

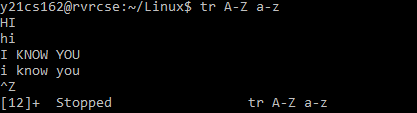


**26 .tr :**

The tr command is a Linux command-line utility that translates or deletes characters from standard input ( stdin ) and writes the result to standard output ( stdout ).

Use tr to perform different text transformations, including case conversion, squeezing or deleting characters, and basic text replacement.

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



**Module – 3**

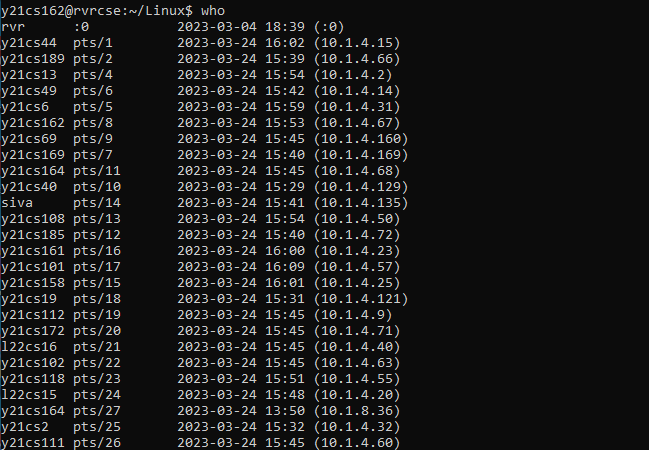
**Disk utilities, Backup and**

**other utilities**

1. **Who :**

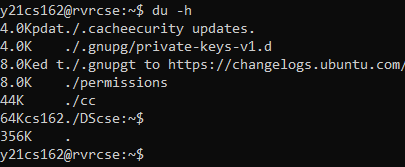
who command is a tool print information about users who are currently logged in.

**Syntax :**who



**2. du:**

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

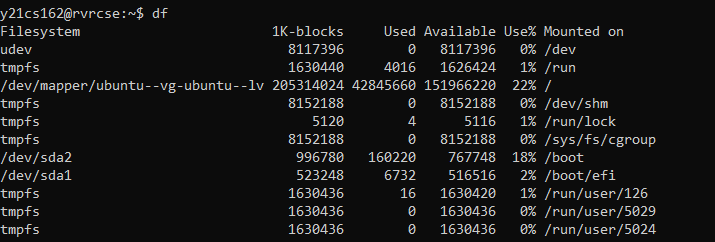


` **3. df :**

It displays the amount of disk space available on the file system containing each file name

argument. If not file name is give, the space available on all currently mounted file systems is

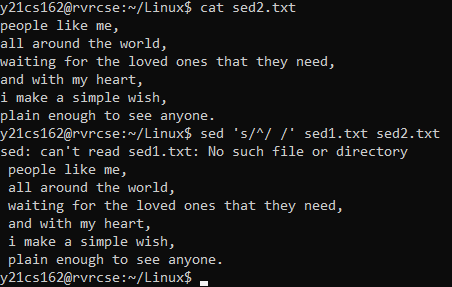
shown.



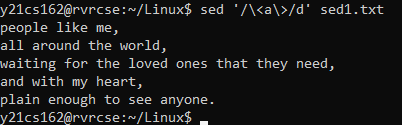
**4. sed :**

The *s*tream *ed*itor utility sed scans one or more files and performs an editing action on all of the lines that match a particular condition.

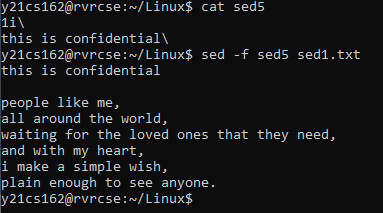
* Substituting text: Substituting first character with a space in each line.



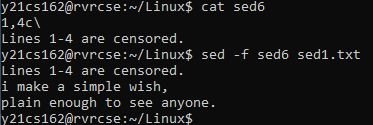
* Deleting text: Deleting only those lines that contain the word ‘a’



* Inserting text:



* Replacing text:

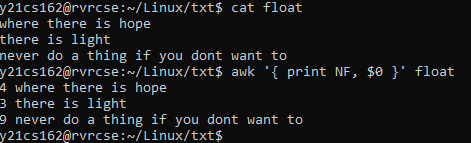


**MODULE – 4**

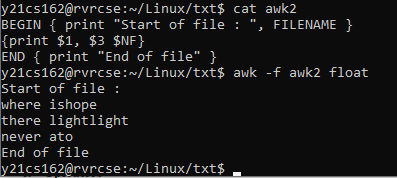
**PROGRAMMABLE TEXT PROCESSING**

***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

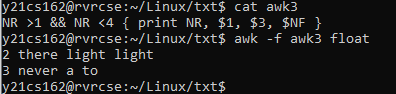
1. **Accessing individual files:**

****

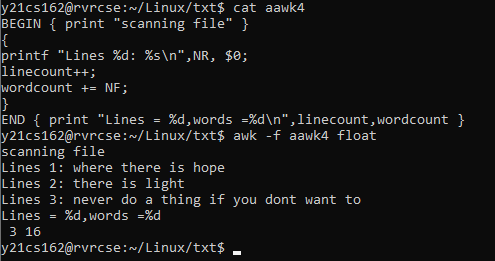
1. **Begin and End :**

****

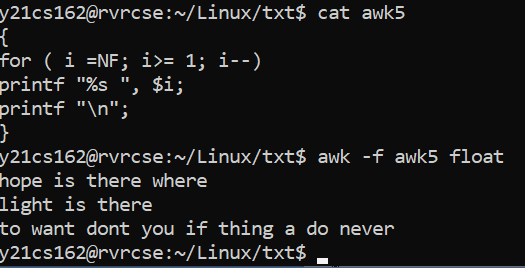
1. **Operators:**

****

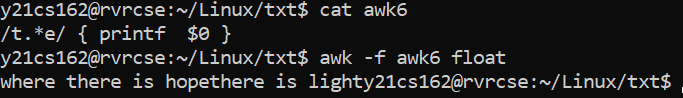
1. **Variables:**

****

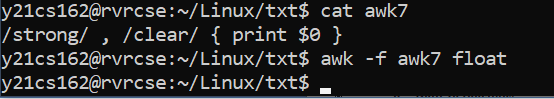
1. **Control Structures:**

****

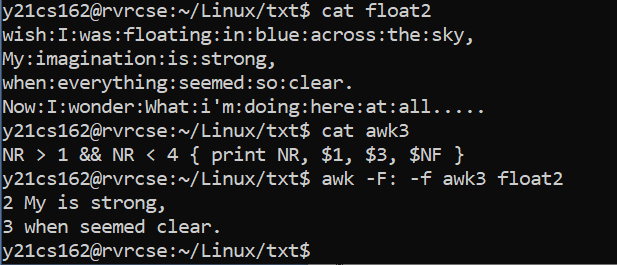
1. **Extended regular expressions:**



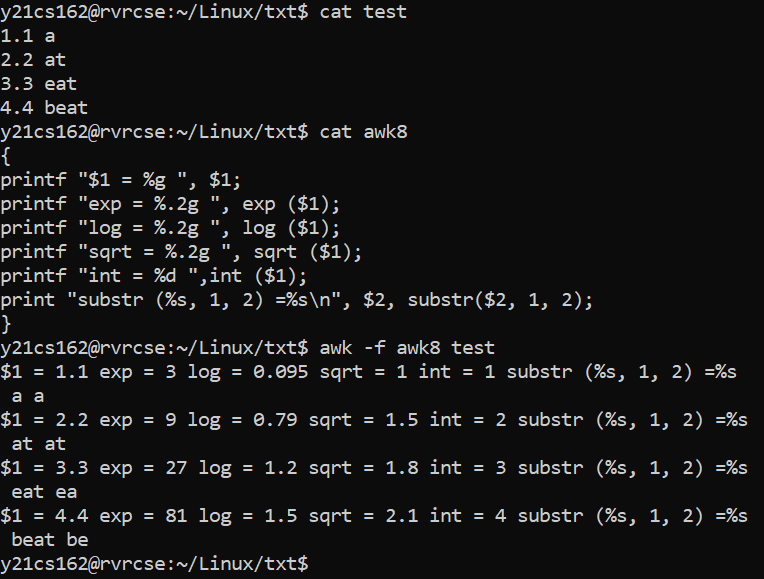
1. **Condition Ranges:**

****

1. **Field Separators:**

****

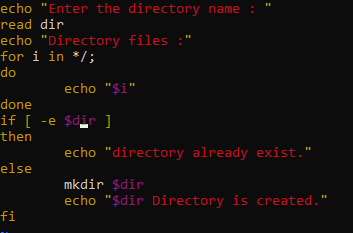
1. **Built-In functions:**

****

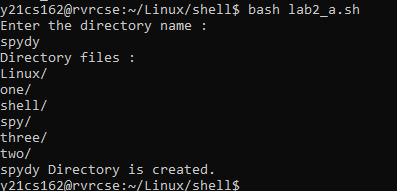
**LAB CYCLE – 2:**

1. To create a directory and list all the directory files in a directory.

SOURCE CODE:

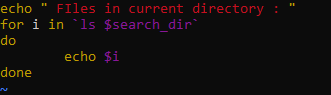


OUTPUT:

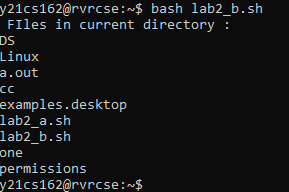


1. To display a list of all the files in the current directory.

SOURCE CODE:

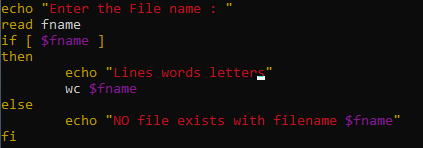


OUTPUT:

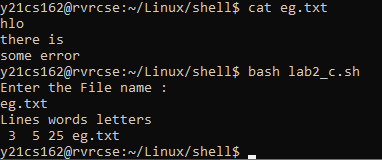


1. To count no of lines, words, and characters of an input file.

SOURCE CODE:

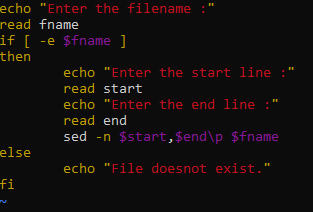


OUTPUT:

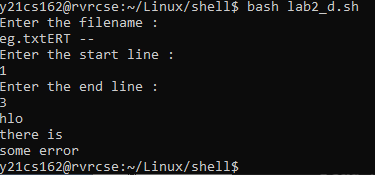


1. To accept a file name starting and ending line numbers as arguments and display all the lines between given line numbers.

SOURCE CODE:

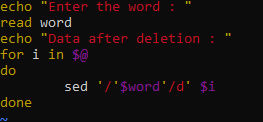


OUTPUT:

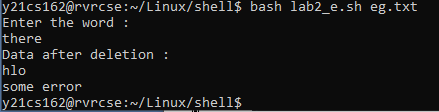


1. To deletes all lines containing the specified word in one or more files supplied as arguments to it.

SOURCE CODE:

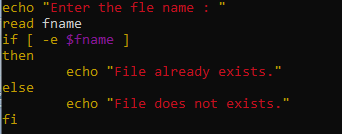


OUTPUT:

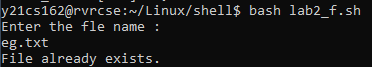


1. To test whether the given file is existing or not.

SOURCE CODE:

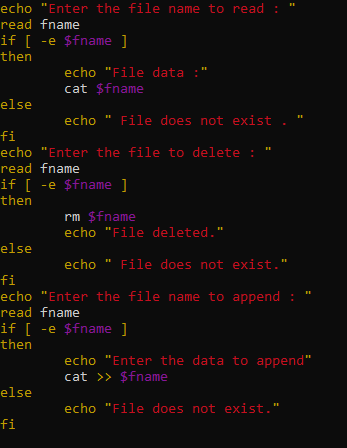


OUTPUT:

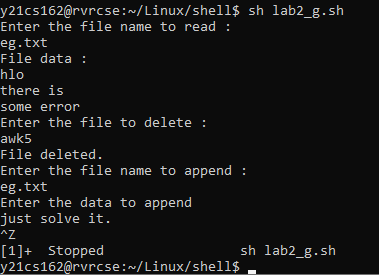


1. To read, delete and append a file.

SOURCE CODE:

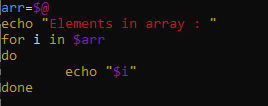


OUTPUT:

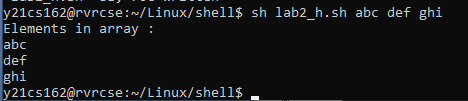


1. To store all command line arguments to an array and print.

SOURCE CODE:

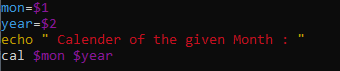


OUTPUT:



1. To print the calender month by default.

SOURCE CODE:



OUTPUT:

