



D Y PATIL INTERNATIONAL UNIVERSITY

**SYSTEMS SOFTWARE
SEMESTER-IV
LAB SHEET - 2**

Name : Suryakant Upadhyay

PRN : 20220802043

Div : A1

1 AIM:

(A) Explain the following commands:

1. clear
2. cal
3. who
4. date
5. mkdir
6. rm
7. cat8. cd
9. cp
10. grep
11. ls
12. mv
- 13.rmdir

2 TOOLS/APPARATUS: Linux operating system.

3 STANDARD PROCEDURES:

3.1 Analyzing the Problem:

- Start the Linux and enter the user name and password.

- Now write startx and after that open the terminal.
- At the terminal try the different commands and see the output.

3.2 Designing the Solution:

- At the terminal first perform the command CAL without and with the different options available for it.
- Like \$ cal and then enter. The calendar will be displayed at the terminal. \$ cal -m and then enter. In the calendar Monday will be displayed as the first day of the week.
- Same way perform the other commands like CLEAR, WHO, DATE, MKDIR, RM etc.

3.3 Implementing the Solution:

3.3.1 Writing Source Code:

1) CAL:

At the terminal write the following:

```
[user1@com]$ cal
```

```
[user1@com]$ cal -m
```

```
[user1@com]$ cal -j
```

```
[user1@com]$ cal -y
```

```
vboxuser@KLinux:~$ cal
February 2024
Su Mo Tu We Th Fr Sa
      1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29

vboxuser@KLinux:~$ cal -m 1
January 2024
Su Mo Tu We Th Fr Sa
      1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

vboxuser@KLinux:~$ cal -j 2
2
January February
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7 32 33 34 35
 8  9 10 11 12 13 14 36 37 38 39 40 41 42
15 16 17 18 19 20 21 43 44 45 46 47 48 49
22 23 24 25 26 27 28 50 51 52 53 54 55 56
29 30 31 57 58 59

March April
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
64 65 66 67 68 69 70 92 93 94 95 96 97 98
71 72 73 74 75 76 77 99 100 101 102 103 104 105
78 79 80 81 82 83 84 106 107 108 109 110 111 112
85 86 87 88 89 90 113 114 115 116 117 118 119
120

May June
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
121 122 123 124 125 126 152 153 154
127 128 129 130 131 132 133 155 156 157 158 159 160 161
134 135 136 137 138 139 140 162 163 164 165 166 167 168
141 142 143 144 145 146 147 169 170 171 172 173 174 175
148 149 150 151 176 177 178 179 180 181

July August
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
183 184 185 186 187 188 189 213 214 215 216 217
190 191 192 193 194 195 196 218 219 220 221 222 223 224
197 198 199 200 201 202 203 225 226 227 228 229 230 231
204 205 206 207 208 209 210 232 233 234 235 236 237 238
211 212 239 240 241 242 243
```

```

          September                October
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
          244 245 274 275 276 277 278 279 280
246 247 248 249 250 251 252 281 282 283 284 285 286 287
253 254 255 256 257 258 259 288 289 290 291 292 293 294
260 261 262 263 264 265 266 295 296 297 298 299 300 301
267 268 269 270 271 272 273 302 303 304

          November                December
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
          305 306 307 308          335 336
309 310 311 312 313 314 315 337 338 339 340 341 342 343
316 317 318 319 320 321 322 344 345 346 347 348 349 350
323 324 325 326 327 328 329 351 352 353 354 355 356 357
330 331 332 333 334          358 359 360 361 362 363 364
vboxuser@KLinux:~$ cal -y 1
          1
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
          1 2 3 4 5          6 7 8 9 10 11 12
2 3 4 5 6 7 8 6 7 8 9 10 11 12 13 14 15 16 17 18 19
9 10 11 12 13 14 15 13 14 15 16 17 18 19 20 21 22 23 24 25 26
16 17 18 19 20 21 22 20 21 22 23 24 25 26 27 28 29 30 31
23 24 25 26 27 28 29 27 28 27 28 29 30 31

          April                May                June
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
          1 2          1 2 3 4 5 6 7          1 2 3 4
3 4 5 6 7 8 9 8 9 10 11 12 13 14 5 6 7 8 9 10 11
10 11 12 13 14 15 16 15 16 17 18 19 20 21 12 13 14 15 16 17 18
17 18 19 20 21 22 23 22 23 24 25 26 27 28 19 20 21 22 23 24 25
24 25 26 27 28 29 30 29 30 31 26 27 28 29 30

          July                August                September
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
          1 2          1 2 3 4 5 6          1 2 3
3 4 5 6 7 8 9 7 8 9 10 11 12 13 4 5 6 7 8 9 10
10 11 12 13 14 15 16 14 15 16 17 18 19 20 11 12 13 14 15 16 17
17 18 19 20 21 22 23 21 22 23 24 25 26 27 18 19 20 21 22 23 24
24 25 26 27 28 29 30 28 29 30 31 25 26 27 28 29 30
31

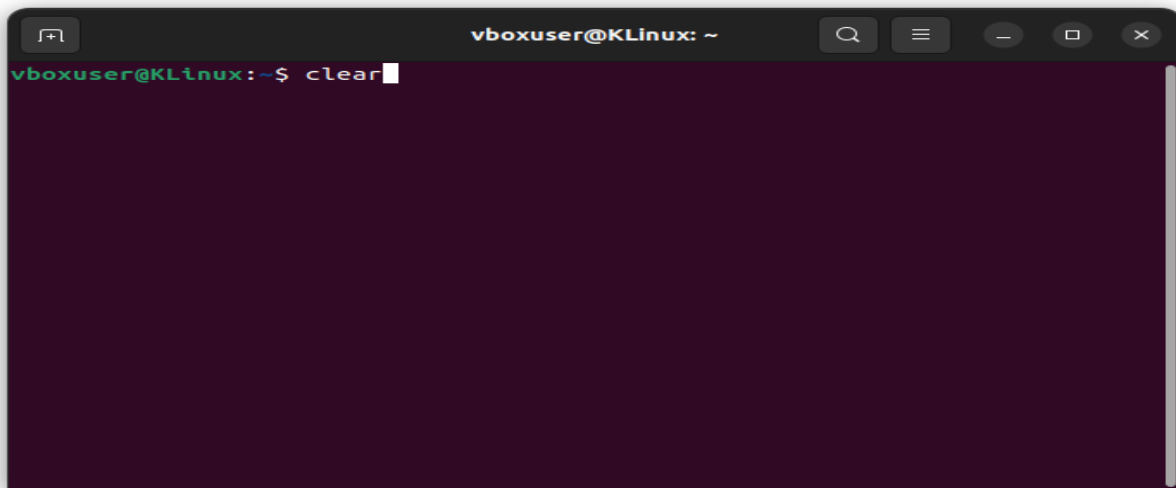
          October                November                December
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
          1          1 2 3 4 5          1 2 3
2 3 4 5 6 7 8 6 7 8 9 10 11 12 4 5 6 7 8 9 10
9 10 11 12 13 14 15 13 14 15 16 17 18 19 11 12 13 14 15 16 17
16 17 18 19 20 21 22 20 21 22 23 24 25 26 18 19 20 21 22 23 24
23 24 25 26 27 28 29 27 28 29 30 25 26 27 28 29 30 31
30 31

```

2) CLEAR:

At the terminal write the following:

```
[user1@com]$ clear
```

A terminal window titled 'vboxuser@KLinux: ~' with a dark purple background. The prompt is 'vboxuser@KLinux:~\$' and the command 'clear' has been entered, resulting in a blank terminal screen.

```
vboxuser@KLinux:~$ clear
```

3) WHO:

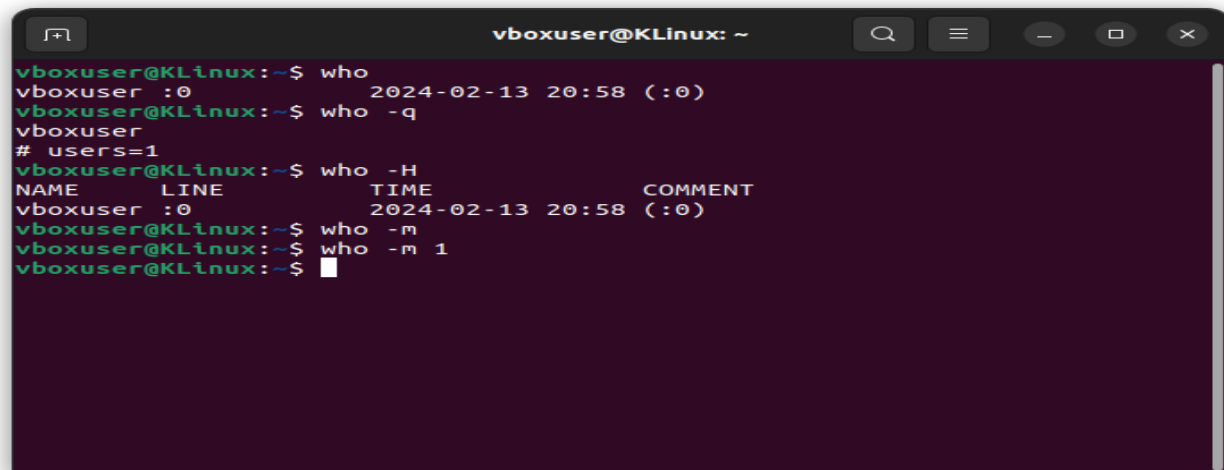
At the terminal write the following:

```
[user1@com]$ who
```

```
[user1@com]$ who -q
```

```
[user1@com]$ who -H
```

```
[user1@com]$ who -m
```

A terminal window titled 'vboxuser@KLinux: ~' with a dark purple background. It shows the output of several 'who' commands. The first command 'who' shows 'vboxuser :0' with a timestamp. The second 'who -q' shows 'vboxuser' and '# users=1'. The third 'who -H' shows a header with columns for NAME, LINE, TIME, and COMMENT. The fourth 'who -m' shows 'vboxuser :0'.

```
vboxuser@KLinux:~$ who
vboxuser :0                2024-02-13 20:58 (:0)
vboxuser@KLinux:~$ who -q
vboxuser
# users=1
vboxuser@KLinux:~$ who -H
NAME      LINE      TIME      COMMENT
vboxuser :0                2024-02-13 20:58 (:0)
vboxuser@KLinux:~$ who -m
vboxuser :0
vboxuser@KLinux:~$ who -m 1
vboxuser@KLinux:~$
```

4) DATE:

At the terminal write the following:

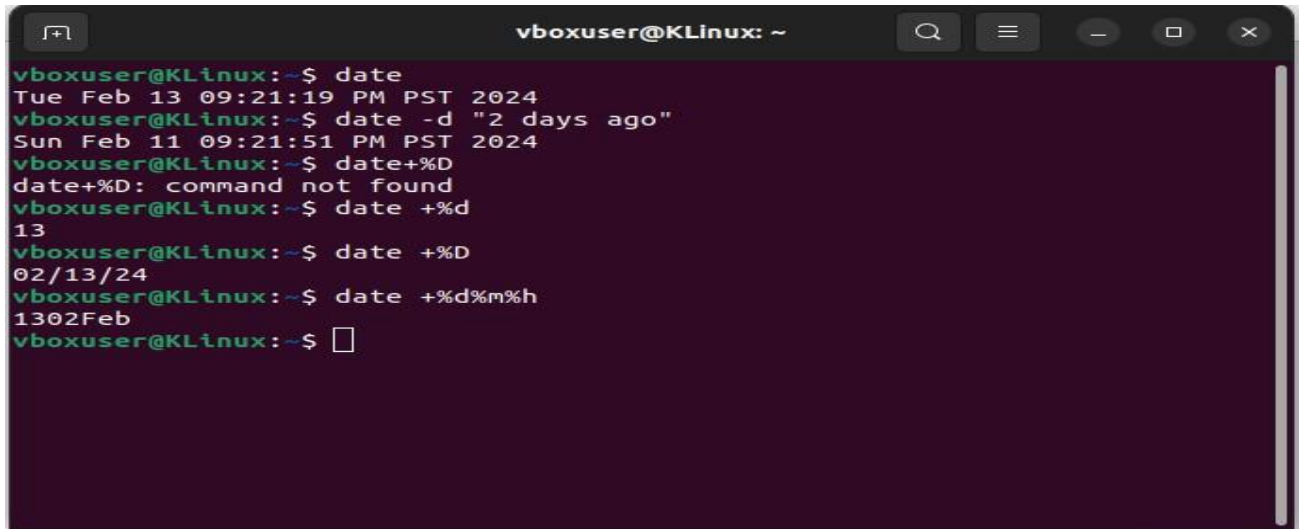
```
[user1@com]$ date
```

```
[user1@com]$ date -d "2 days ago"
```

```
[user1@com]$ date +%D
```

```
[user1@com]$ date +%d
```

[user1@com]\$ date +%d%m%h

A terminal window titled 'vboxuser@KLinux: ~' with standard window controls. The terminal shows a series of commands and their outputs: 'date' returns the current date and time; 'date -d "2 days ago"' returns the date two days prior; 'date +%D' returns the date in MM/DD/YY format; 'date +%d' returns the day of the month; 'date +%D' returns the full date; and 'date +%d%m%h' returns the date and time in ddmmhh format.

```
vboxuser@KLinux:~$ date
Tue Feb 13 09:21:19 PM PST 2024
vboxuser@KLinux:~$ date -d "2 days ago"
Sun Feb 11 09:21:51 PM PST 2024
vboxuser@KLinux:~$ date +%D
date +%D: command not found
vboxuser@KLinux:~$ date +%d
13
vboxuser@KLinux:~$ date +%D
02/13/24
vboxuser@KLinux:~$ date +%d%m%h
1302Feb
vboxuser@KLinux:~$
```

5) MKDIR and RM:

At the terminal write the following:

[user1@com]\$ cd Desktop/

[user1@com]\$ cd newfiles/

[user1@com]\$ mkdir newfile1

[user1@com]\$ rm Sum_Of_Digits.txt

[user1@com]\$ ls

A terminal window titled 'vboxuser@KLinux: ~/Desktop/newfiles' with standard window controls. The terminal shows a sequence of commands: 'cd Desktop' to move to the desktop, 'mkdir newfiles' to create a new directory, 'ls' to list files, 'cd newfiles/' to move into the new directory, 'mkdir newfile1' to create another directory, 'ls' to list files, 'touch Sum_Of_Digits.txt' to create a file, 'ls' to list files, 'rm Sum_Of_Digits.txt' to remove the file, 'ls' to list files, and finally 'cd' to return to the previous directory.

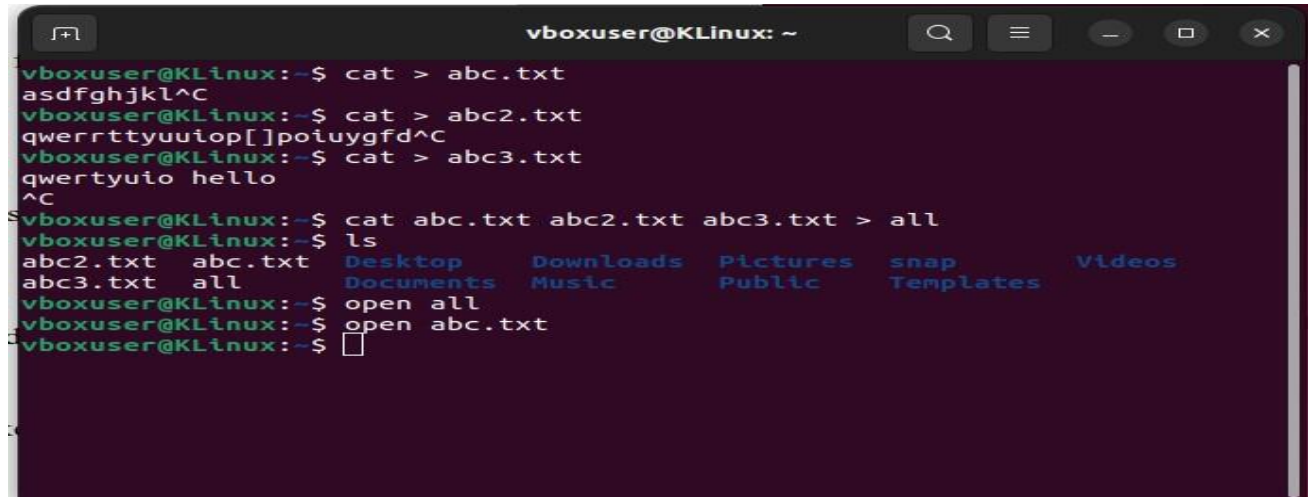
```
vboxuser@KLinux:~$ cd Desktop
vboxuser@KLinux:~/Desktop$ mkdir newfiles
vboxuser@KLinux:~/Desktop$ ls
newfiles  'SS LAB Assignment 2.pdf'
vboxuser@KLinux:~/Desktop$ cd newfiles/
vboxuser@KLinux:~/Desktop/newfiles$ mkdir newfile1
vboxuser@KLinux:~/Desktop/newfiles$ ls
newfile1
vboxuser@KLinux:~/Desktop/newfiles$ touch Sum_Of_Digits.txt
vboxuser@KLinux:~/Desktop/newfiles$ ls
newfile1  Sum_Of_Digits.txt
vboxuser@KLinux:~/Desktop/newfiles$ rm Sum_Of_Digits.txt
vboxuser@KLinux:~/Desktop/newfiles$ ls
newfile1
vboxuser@KLinux:~/Desktop/newfiles$
```

6) **cat** *cat* allows you to read multiple files and then print them out. You can combine files by using the > operator and append files by using >>.

Syntax: **cat** [argument] [specific file]

Example: *cat abc.txt*

If you want to append three files (abc.txt, def.txt, xyz.txt), give the command as, *cat abc.txt def.txt xyz.txt > all*

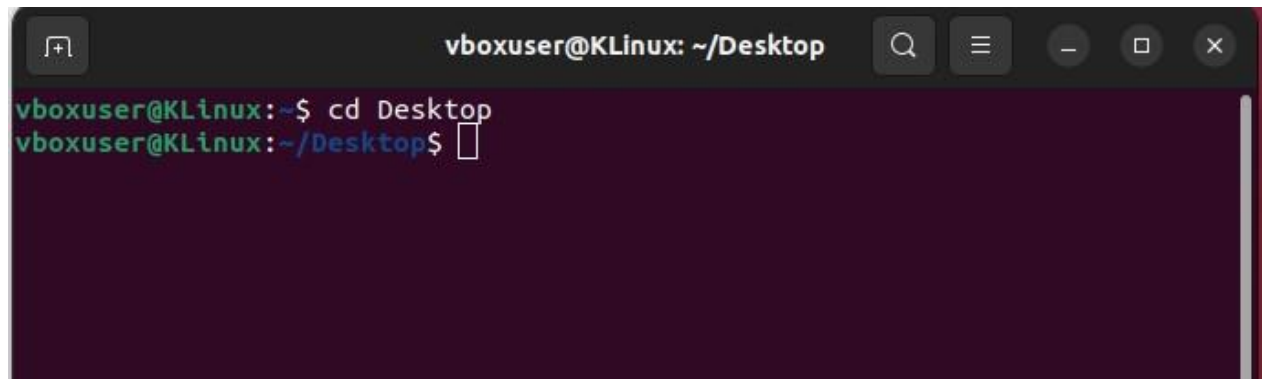
A terminal window titled 'vboxuser@KLinux: ~' with standard window controls. The user enters several commands: 'cat > abc.txt' followed by 'asdfghjkl^C', 'cat > abc2.txt' followed by 'qwerrttyuuiop[]poiuygfd^C', and 'cat > abc3.txt' followed by 'qwertyuio hello ^C'. Then, they run 'cat abc.txt abc2.txt abc3.txt > all'. After that, they run 'ls', which shows a directory listing with files 'abc2.txt', 'abc.txt', and 'abc3.txt', and directories 'Desktop', 'Downloads', 'Pictures', 'snap', and 'Videos'. Finally, they run 'open all' and 'open abc.txt', with the latter opening a file editor window (represented by a small icon).

7) cd, chdir

cd (or *chdir*) stands for “change directory”. This command is the key command to move around your file structure.

Syntax: *cd [name of directory you want to move to]*

When changing directories, start with / and then type the complete file path, like *cd /vvs/abc/xyz*

A terminal window titled 'vboxuser@KLinux: ~/Desktop' with standard window controls. The user enters 'cd Desktop' and then the prompt changes to 'vboxuser@KLinux: ~/Desktop\$'.

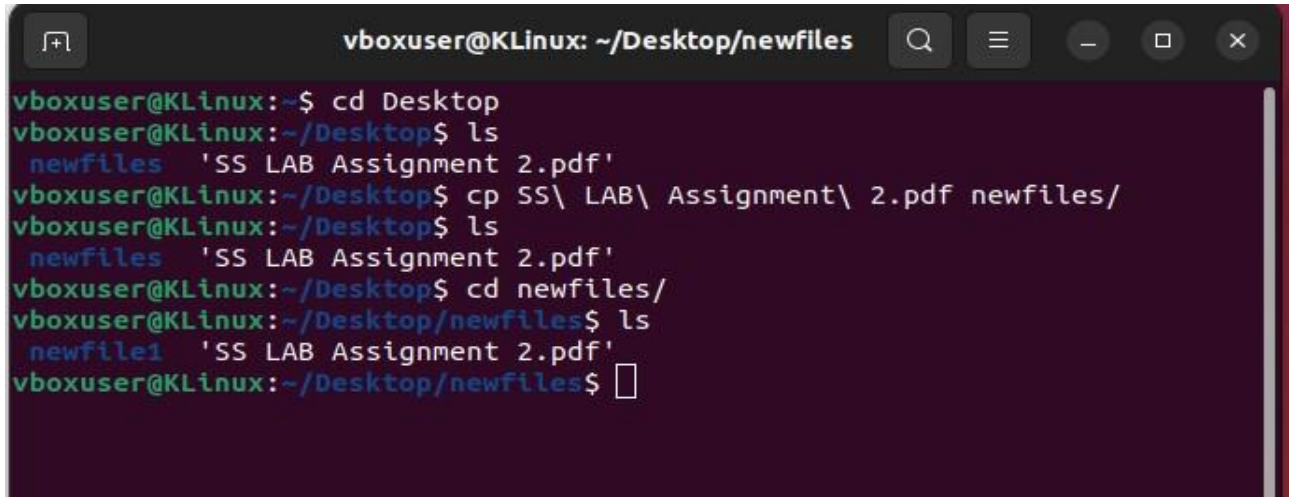
8) cp

The *cp* command copies files or directories from one place to another. You can copy a set of files to another file, or copy one or more files under the same name in a directory. If the destination of the file you want to copy is an existing file, then the existing file is overwritten. If the destination is an existing directory, then the file is copied into that directory. Syntax: *cp [options] file1 file2*

If you want to copy the file *favourites.html* into the directory called *laksh*, you give the command as:

cp favourites.html /vvs/laksh/

A handy option to use with ***cp*** is ***-r***. This recursively copies a particular directory and all of its contents to the specified directory, so you won't have to copy one file at a time.

A terminal window titled 'vboxuser@KLinux: ~/Desktop/newfiles' with standard window controls. The terminal shows the following commands and output:

```
vboxuser@KLinux:~$ cd Desktop
vboxuser@KLinux:~/Desktop$ ls
newfiles  'SS LAB Assignment 2.pdf'
vboxuser@KLinux:~/Desktop$ cp SS\ LAB\ Assignment\ 2.pdf newfiles/
vboxuser@KLinux:~/Desktop$ ls
newfiles  'SS LAB Assignment 2.pdf'
vboxuser@KLinux:~/Desktop$ cd newfiles/
vboxuser@KLinux:~/Desktop/newfiles$ ls
newfile1  'SS LAB Assignment 2.pdf'
vboxuser@KLinux:~/Desktop/newfiles$
```

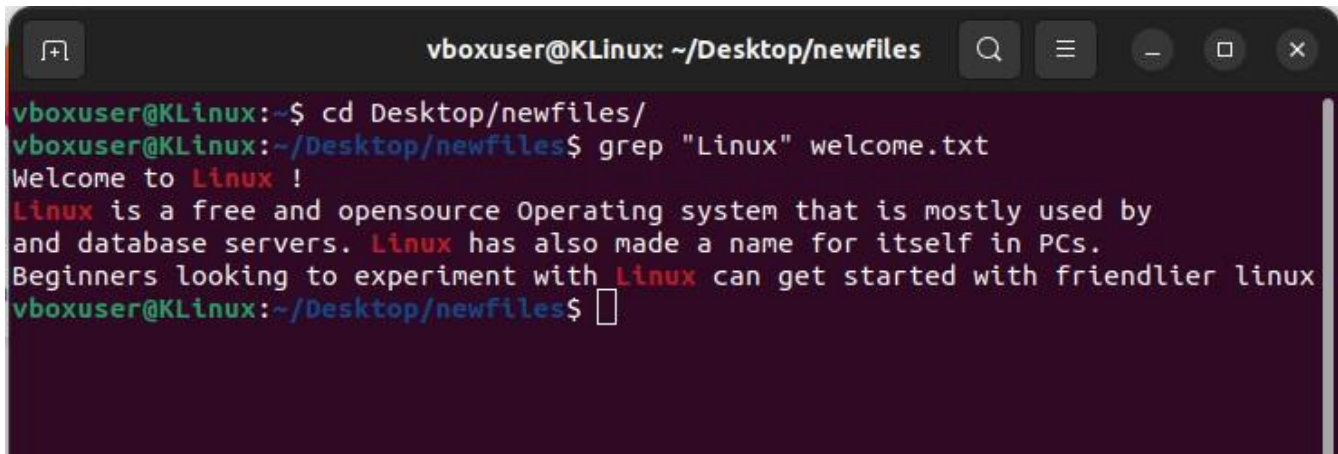
9) ***grep***

The ***grep*** command searches a file or files for lines that match a provided regular expression (“grep” comes from a command meaning to **g**lobally search for a **r**egular **e**xpression and then **p**rint the found matches). Syntax: ***grep [options] regular expression [files]***

To exit this command, type 0 if lines have matched, 1 if no lines match, and 2 for errors. This is very useful if you need to match things in several files. If you wanted to find out which files in our ***vvs*** directory contained the word “***mca***” you could use ***grep*** to search the directory and match those files with that word. All that you have to do is give the command as shown:

grep 'mca' /vvs/*

The ******* used in this example is called a meta-character, and it represents matching zero or more of the preceding characters. In this example, it is used to mean “all files and directories in this directory”. So, ***grep*** will search all the files and directories in ***vvs*** and tell you which files contain “***mca***”.

A terminal window titled 'vboxuser@KLinux: ~/Desktop/newfiles'. The prompt is 'vboxuser@KLinux:~\$'. The user enters 'cd Desktop/newfiles/'. The prompt changes to 'vboxuser@KLinux:~/Desktop/newfiles\$'. The user enters 'grep "Linux" welcome.txt'. The output is 'Welcome to Linux !' followed by a paragraph of text about Linux: 'Linux is a free and opensource Operating system that is mostly used by and database servers. Linux has also made a name for itself in PCs. Beginners looking to experiment with Linux can get started with friendlier linux'. The prompt returns to 'vboxuser@KLinux:~/Desktop/newfiles\$' with a cursor.

10) ls

ls will list all the files in the current directory. If one or more files are given, *ls* will display the files contained within “name” or list all the files with the same name as “name”. The files can be displayed in a variety of formats using various options.

Syntax: *ls [options] [names]* *ls* is a command you'll end up using all the time. It simply stands for list. If you are in a directory and you want to know what files and directories are inside that directory, type *ls*. Sometimes the list of files is very long and it flies past your screen so quickly you miss the file you want. To overcome this problem give the command as shown below: *ls /more*

The character | (called pipe) is typed by using shift and the \ key. | *more* will show as many files as will fit on your screen, and then display a highlighted “*more*” at the bottom. If you want to see the next screen, hit enter (for moving one line at a time) or the spacebar (to move a screen at a time). / *more* can be used anytime you wish to view the output of a command in this way.

A useful option to use with *ls* command is *-l*. This will list the files and directories in a long format. This means it will display the permissions (see *chmod*), owners, group, size, date and time the file was last modified, and the filename. *drwxrwxr-xvvs staff 512 Apr 5 09:34 sridhar.txt -rwx-rw-r- vvs staff 4233 Apr 1 10:20 resume.txt*

-rwx-r--r-- vvs staff 4122 Apr 1 12:01 favourites.html

There are several other options that can be used to modify the *ls* command, and many of these options can be combined. *-a* will list all files in a directory, including those files normally hidden. *-F* will flag filenames by putting / on directories, @ on symbolic links, and * on executable files.

```
vboxuser@KLinux: ~  
vboxuser@KLinux:~$ ls  
abc2.txt  abc.txt  Desktop  Downloads  Pictures  snap  Videos  
abc3.txt  all      Documents Music      Public   Templates  
vboxuser@KLinux:~$
```

11) **mv** *mv* moves files and directories. It can also be used to rename files or directories.

Syntax: **mv** [options] source target

If you wanted to rename vvs.txt to vsv.txt, you should give the command as:

```
mv vvs.txt vsv.txt
```

After executing this command, vvs.txt would no longer exist, but a file with name vsv.txt would now exist with the same contents.

```
vboxuser@KLinux:~/Documents$ ls  
hello.txt  new.txt  
vboxuser@KLinux:~/Documents$ mv new.txt new2.txt  
vboxuser@KLinux:~/Documents$ ls  
hello.txt  new2.txt
```

12) **rm** *rm* removes or deletes files from a directory. Syntax: **rm** [options] files

In order to remove a file, you must have write permission to the directory where the file is located. While removing a which doesn't have write permission on, a prompt will come up asking you whether or not you wish to override the write protection.

The **-r** option is very handy and very dangerous. **-r** can be used to remove a directory and all its contents. If you use the **-i** option, you can possibly catch some disastrous mistakes because it'll ask you to confirm whether you really want to remove a file before going ahead and doing it.

13) **rmdir** *rmdir* allows you to remove or delete directories but not their contents. A directory must be empty in order to remove it using this command.

Syntax: **rmdir** [options] directories

If you wish to remove a directory and all its contents, you should use **rm -r**.

```
vboxuser@KLinux:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
vboxuser@KLinux:~$ cd Documents
vboxuser@KLinux:~/Documents$ mkdir new
vboxuser@KLinux:~/Documents$ ls
new
vboxuser@KLinux:~/Documents$ rmdir new
vboxuser@KLinux:~/Documents$ ls
vboxuser@KLinux:~/Documents$ touch hello.html
vboxuser@KLinux:~/Documents$ ls
hello.html
vboxuser@KLinux:~/Documents$ rm hello.html
vboxuser@KLinux:~/Documents$ ls
```

Conclusion: We try the different commands on terminal on linux and analysis the output.