SYSTEM SOFTWARE
SEMESTER- IV
LAB SHEET- 02

Name: Mohammad Ahmad Ansari
PRN: 20220802059
Batch: A2

1) **AIM:**
   (A) Explain the following commands:
   a) clear: Clears the terminal screen, removing any text currently displayed.
   b) Cal: Displays a calender for the current month or specified month and year.
   c) who: Lists currently logged-in users on the system, including their usernames, terminals, and idle times.
   d) date: Displays the current date and time in various formats depending on options.
   e) mkdir: Creates a new directory.
   f) rm: Removes files or directories.
   g) cat: Displays the contents of a file on the terminal.
   h) cd: Changes the current working directory.
   i) cp: Copies files or directories. Specify source and destination paths.
   j) grep: Searches text files for lines matching a specified pattern.
   k) ls: Lists files and directories in the current working directory
   l) mv: Moves or renames files or directories.
   m) rmdir: Removes an empty directories.
2) **TOOLS/APPARTUS:** 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 differentoptions 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:**
Writing Source Code:

1) **CAL:**

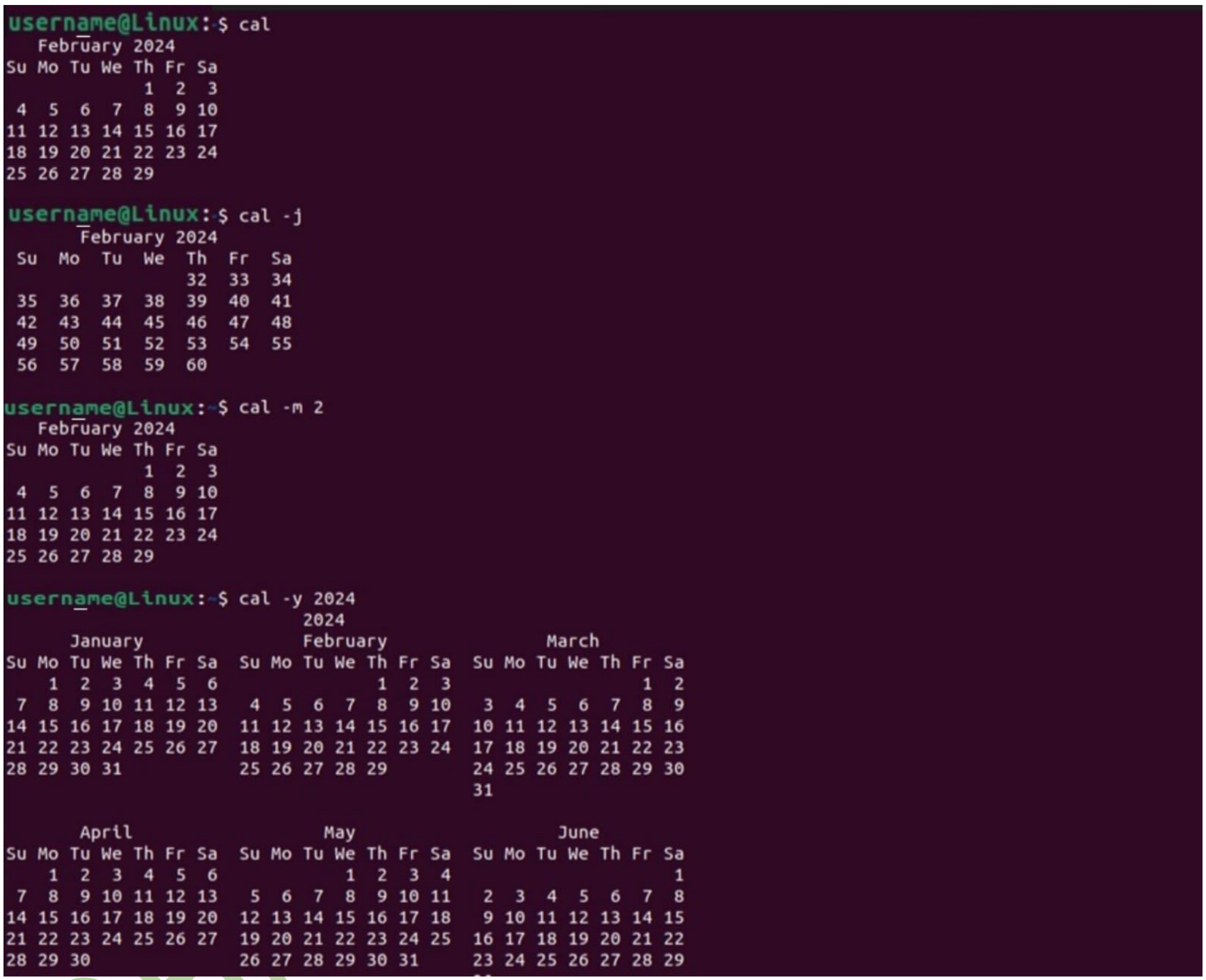
At the terminal write the following

[user1@com]$ cal

[user1@com]$ cal -m

[user1@com]$ cal -j

[user1@com]$ cal -y



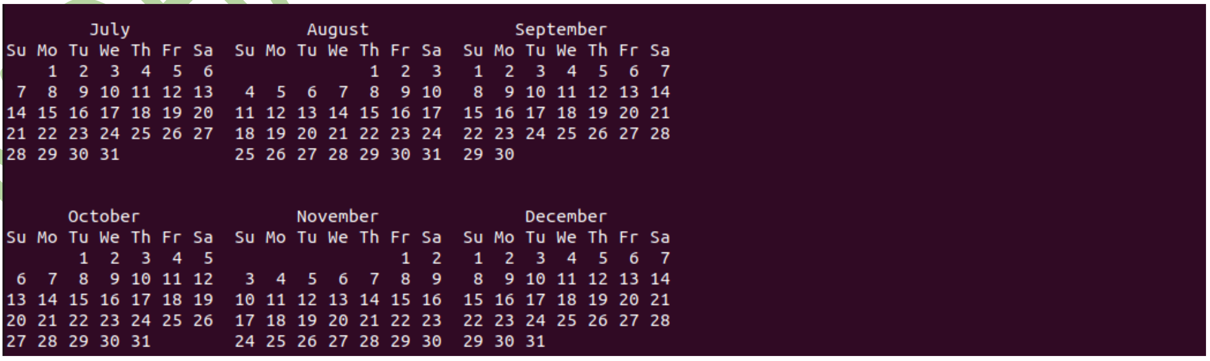
```
username@Linux:~$ cal
   February 2024
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29

username@Linux:~$ cal -j
       February 2024
 Su  Mo  Tu  We  Th  Fr  Sa
                 32  33  34
 35  36  37  38  39  40  41
 42  43  44  45  46  47  48
 49  50  51  52  53  54  55
 56  57  58  59  60

username@Linux:~$ cal -m 2
   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

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



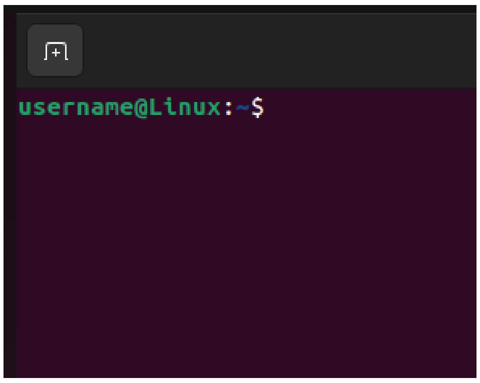
```
        July                 August              September
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6               1  2  3   1  2  3  4  5  6  7
 7  8  9 10 11 12 13   4  5  6  7  8  9 10   8  9 10 11 12 13 14
14 15 16 17 18 19 20  11 12 13 14 15 16 17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


      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  2  3  4  5                  1  2   1  2  3  4  5  6  7
 6  7  8  9 10 11 12   3  4  5  6  7  8  9   8  9 10 11 12 13 14
13 14 15 16 17 18 19  10 11 12 13 14 15 16  15 16 17 18 19 20 21
20 21 22 23 24 25 26  17 18 19 20 21 22 23  22 23 24 25 26 27 28
27 28 29 30 31        24 25 26 27 28 29 30  29 30 31
```

2) CLEAR:

At the terminal write the following:

[user1@com]$ clear



```
username@Linux:~$
```

3) WHO:

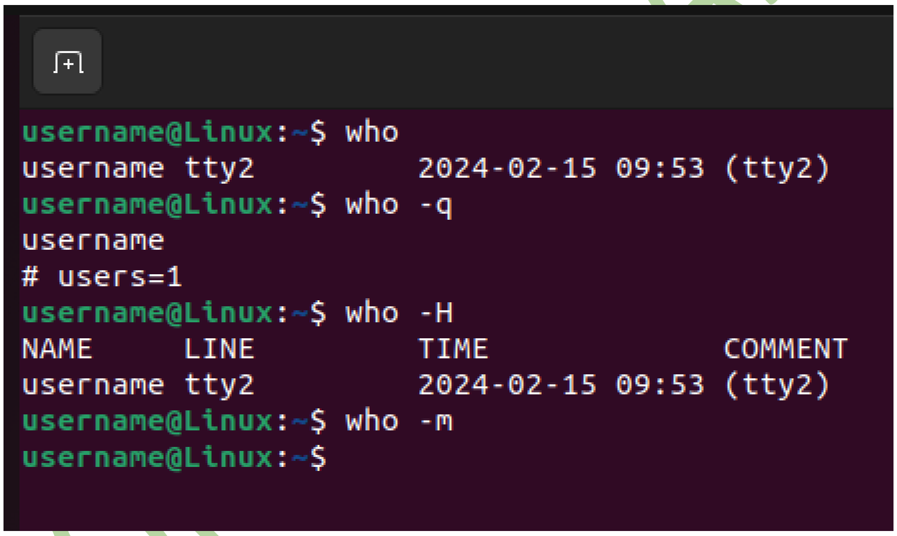
At the terminal write the following:

[user1@com]$ who

[user1@com]$ who -q

[user1@com]$ who -H

[user1@com]$ who -m



```
username@Linux:~$ who
username tty2         2024-02-15 09:53 (tty2)
username@Linux:~$ who -q
username
# users=1
username@Linux:~$ who -H
NAME     LINE         TIME             COMMENT
username tty2         2024-02-15 09:53 (tty2)
username@Linux:~$ who -m
username@Linux:~$
```

4) **DATE:**

At the terminal write the following:

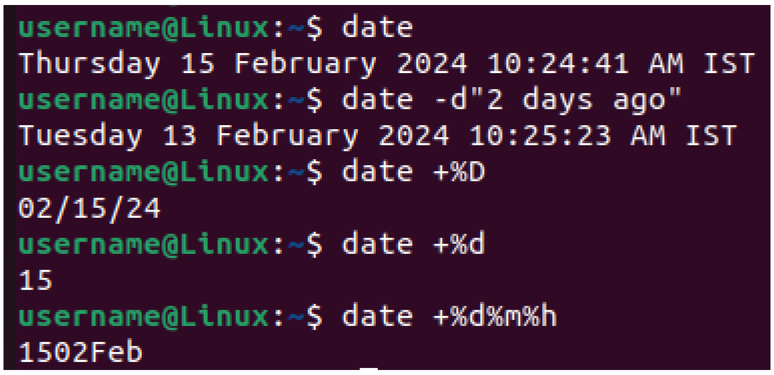
[user1@com]$ date

[user1@com]$ date –d “2 days ago”

[user1@com]$ date +%D

[user1@com]$ date +%d

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



```
username@Linux:~$ date
Thursday 15 February 2024 10:24:41 AM IST
username@Linux:~$ date -d"2 days ago"
Tuesday 13 February 2024 10:25:23 AM IST
username@Linux:~$ date +%D
02/15/24
username@Linux:~$ date +%d
15
username@Linux:~$ date +%d%m%h
1502Feb
```

5) **MKDIR and RM:**

At the terminal write the following:

[user1@com]$ cd Desktop/

[user1@com]$ ls

[user1@com]$ cd newfiles/

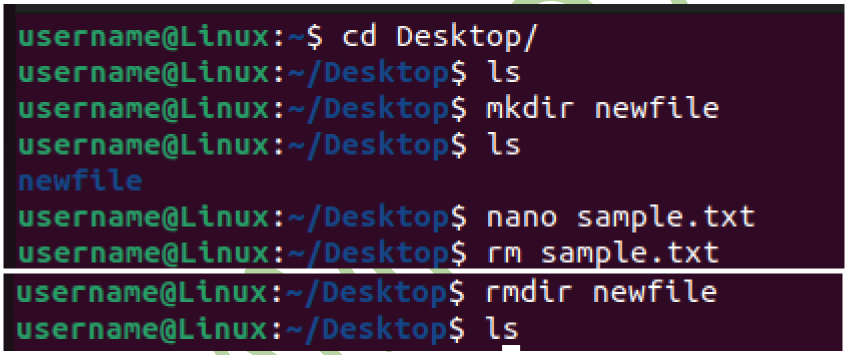
[user1@com]$ ls

[user1@com]$ mkdir newfile1

[user1@com]$ ls

[user1@com]$ rm Sum_Of_Digits.txt

[user1@com]$ ls



```
username@Linux:~$ cd Desktop/
username@Linux:~/Desktop$ ls
username@Linux:~/Desktop$ mkdir newfile
username@Linux:~/Desktop$ ls
newfile
username@Linux:~/Desktop$ nano sample.txt
username@Linux:~/Desktop$ rm sample.txt
username@Linux:~/Desktop$ rmdir newfile
username@Linux:~/Desktop$ ls
```

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 >>.

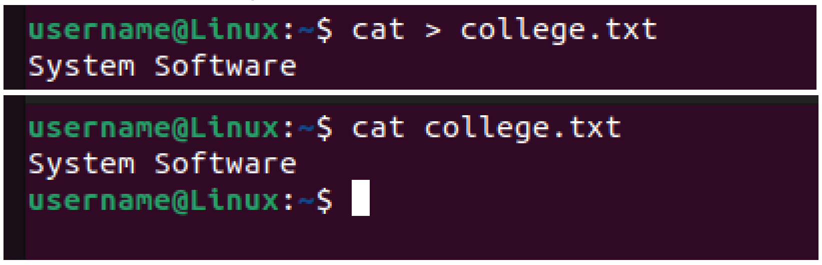
Syntax: cat [argument] [specific file]

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

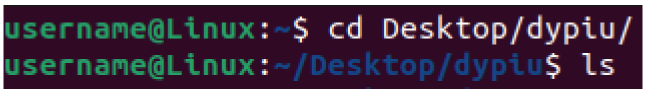


```
username@Linux:~$ cat > college.txt
System Software
username@Linux:~$ cat college.txt
System Software
username@Linux:~$
```

7) **cd, chdir**

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

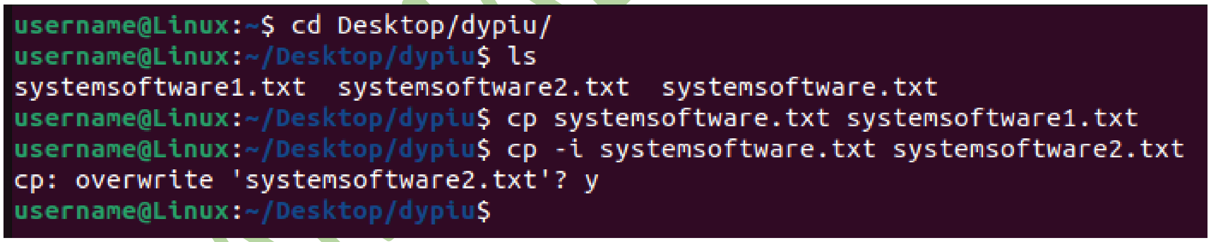
Syntax: cd [name of directory you want to move to]
When changing directories, start with / and then type the complete file path, like cd /vvs/abc/xyz



```
username@Linux:~$ cd Desktop/dypiu/
username@Linux:~/Desktop/dypiu$ ls
```

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 in a directory. If the destination is an existing directory, then the file is copied into that directory.
Syntax: cp[options] file1 file 2
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.

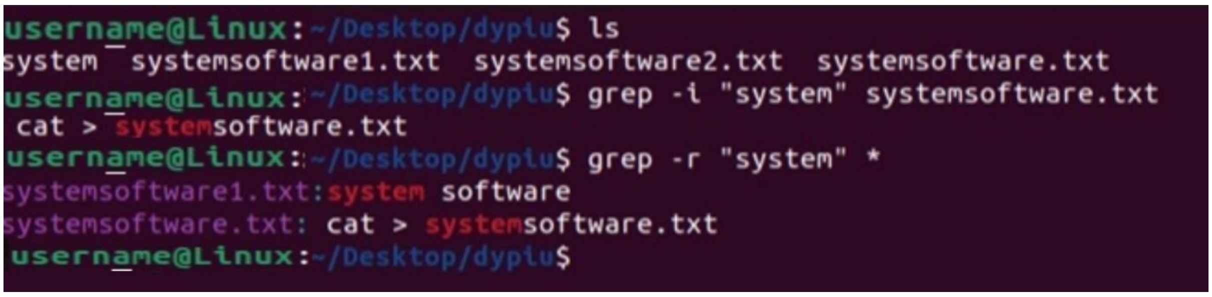


```
username@Linux:~$ cd Desktop/dypiu/
username@Linux:~/Desktop/dypiu$ ls
systemsoftware1.txt  systemsoftware2.txt  systemsoftware.txt
username@Linux:~/Desktop/dypiu$ cp systemsoftware.txt systemsoftware1.txt
username@Linux:~/Desktop/dypiu$ cp -i systemsoftware.txt systemsoftware2.txt
cp: overwrite 'systemsoftware2.txt'? y
username@Linux:~/Desktop/dypiu$
```

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 tofind 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 vvsand tell you which files contain "mca".



```
username@Linux:~/Desktop/dypiu$ ls
system  systemsoftware1.txt  systemsoftware2.txt  systemsoftware.txt
username@Linux:~/Desktop/dypiu$ grep -i "system" systemsoftware.txt
 cat > systemsoftware.txt
username@Linux:~/Desktop/dypiu$ grep -r "system" *
systemsoftware1.txt:system software
systemsoftware.txt: cat > systemsoftware.txt
username@Linux:~/Desktop/dypiu$
```

10) **ls**

lswill list all the files in the current directory. If one or more files are given, ***ls*** willdisplay 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 ina 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 quicklyyou miss the file you want. To overcome this problem give the command as shown below:

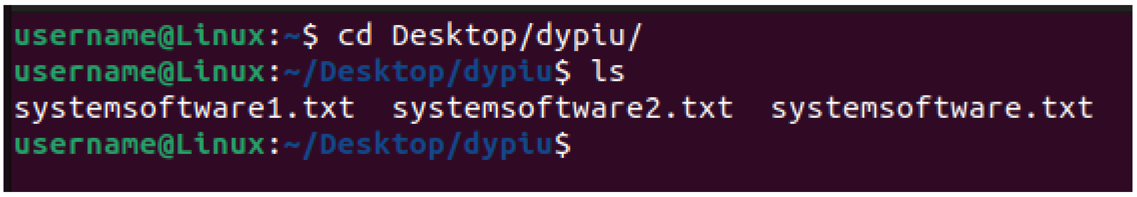
ls | more

The character | (called pipe) is typed by using shift and the \ key. | ***more*** will show asmany 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) orthe 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

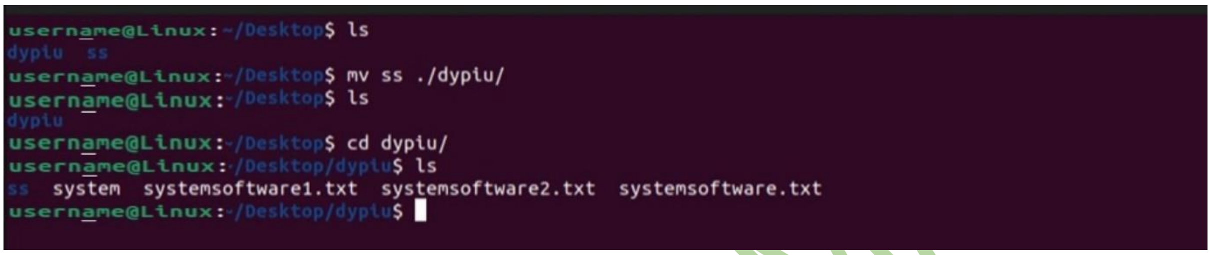
There are several other options that can be used to modify the ls command, and manyof these options can be combined. ***-a*** will list all files in a directory, including those files normally hidden. ***-F*** will flag filenames by putting / on directories, @ on symbolic links, and * on executable files.



```
username@Linux:~$ cd Desktop/dypiu/
username@Linux:~/Desktop/dypiu$ ls
systemsoftware1.txt  systemsoftware2.txt  systemsoftware.txt
username@Linux:~/Desktop/dypiu$
```

11) **mv**

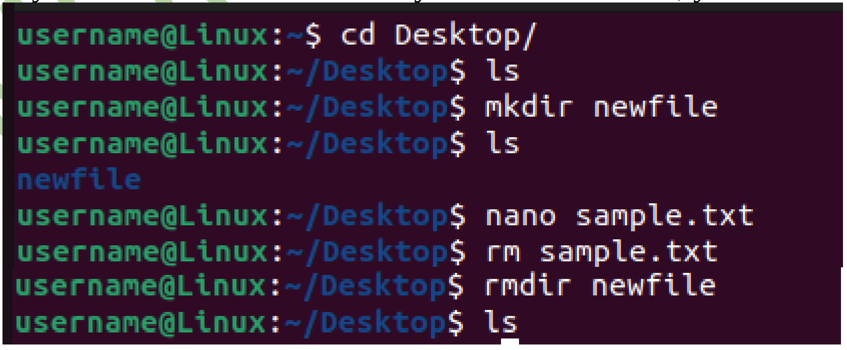
mv moves files and directories. It can also be used to rename files or directories.
Syntax: mv [options] source target
If you wanted to rename vvs.txt to vsv.txt, you should give the command as:mv vvs.txt vsv.txt
After executing this command, vvs.txt would no longer exist, but a file with namevsv.txt would now exist with the same contents.



```
username@Linux:~/Desktop$ ls
dypiu  ss
username@Linux:~/Desktop$ mv ss ./dypiu/
username@Linux:~/Desktop$ ls
dypiu
username@Linux:~/Desktop$ cd dypiu/
username@Linux:~/Desktop/dypiu$ ls
ss  system  systemsoftware1.txt  systemsoftware2.txt  systemsoftware.txt
username@Linux:~/Desktop/dypiu$ 
```

12) **rm and rmdir**

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 thefile is located. While removing a which does'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 directoryand 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.
rmdir allows you to remove or delete directories but not their contents. A directorymust 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.



```
username@Linux:~$ cd Desktop/
username@Linux:~/Desktop$ ls
username@Linux:~/Desktop$ mkdir newfile
username@Linux:~/Desktop$ ls
newfile
username@Linux:~/Desktop$ nano sample.txt
username@Linux:~/Desktop$ rm sample.txt
username@Linux:~/Desktop$ rmdir newfile
username@Linux:~/Desktop$ ls
```