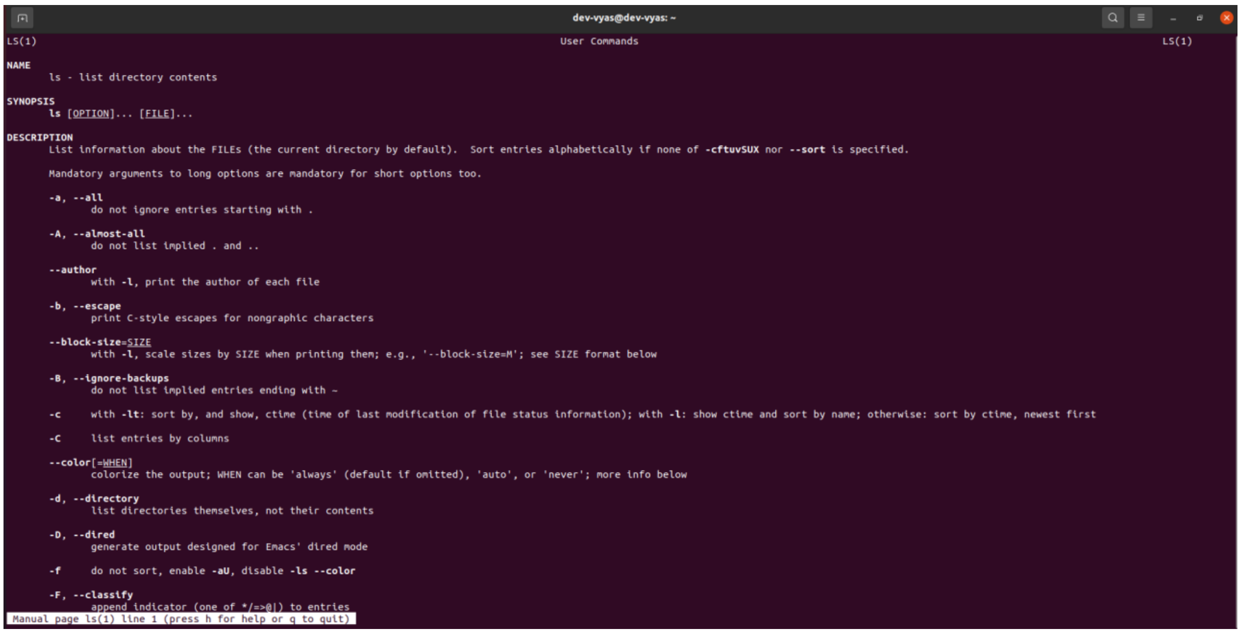
Practical-1

Practical:- Study of Basic commands of Linux/UNIX.

- **man command:-** It is the interface used to view the system's reference manuals.
  Syntax:- man [command name]

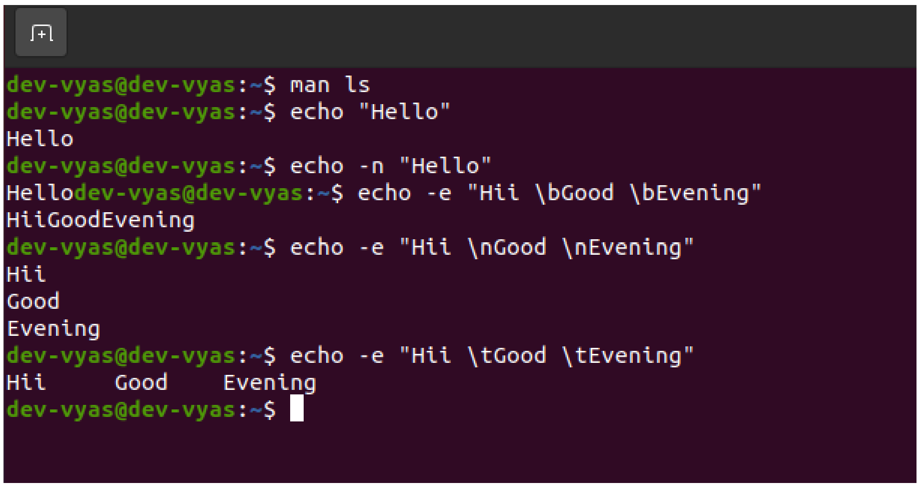


```
LS(1)                                   User Commands                                   LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List information about the FILEs (the current directory by default).  Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
              with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below

       -B, --ignore-backups
              do not list implied entries ending with ~

       -c     with -lt: sort by, and show, ctime (time of last modification of file status information); with -l: show ctime and sort by name; otherwise: sort by ctime, newest first

       -C     list entries by columns

       --color[=WHEN]
              colorize the output; WHEN can be 'always' (default if omitted), 'auto', or 'never'; more info below

       -d, --directory
              list directories themselves, not their contents

       -D, --dired
              generate output designed for Emacs' dired mode

       -f     do not sort, enable -aU, disable -ls --color

       -F, --classify
              append indicator (one of */=>@|) to entries
 Manual page ls(1) line 1 (press h for help or q to quit)
```

- **echo command:-** Display a line of text/string on standard output or a file.
  Syntax:- echo [option] [string]

Option	**Use**
echo -n	Do not output a trailing newline
echo -e	Enable interpretation of backslash escape sequences

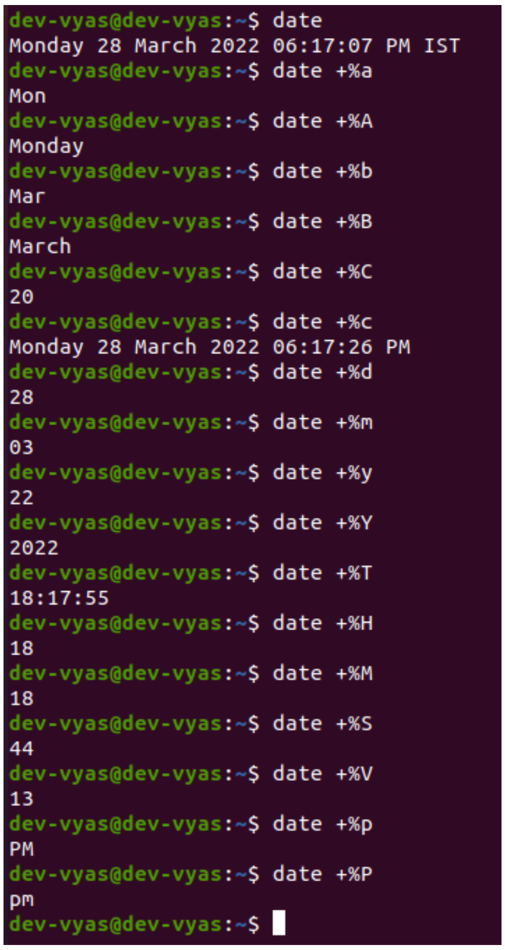
\b	It removes all the spaces in between the text
\n	It creates new line from where it is used
\t	It create horizontal tab spaces



```
dev-vyas@dev-vyas:~$ man ls
dev-vyas@dev-vyas:~$ echo "Hello"
Hello
dev-vyas@dev-vyas:~$ echo -n "Hello"
Hellodev-vyas@dev-vyas:~$ echo -e "Hii \bGood \bEvening"
HiiGoodEvening
dev-vyas@dev-vyas:~$ echo -e "Hii \nGood \nEvening"
Hii
Good
Evening
dev-vyas@dev-vyas:~$ echo -e "Hii \tGood \tEvening"
Hii     Good    Evening
dev-vyas@dev-vyas:~$ 
```

- **date command:- :** Print or set the system date and time.
  Syntax:- date [OPTION]... [+FORMAT]

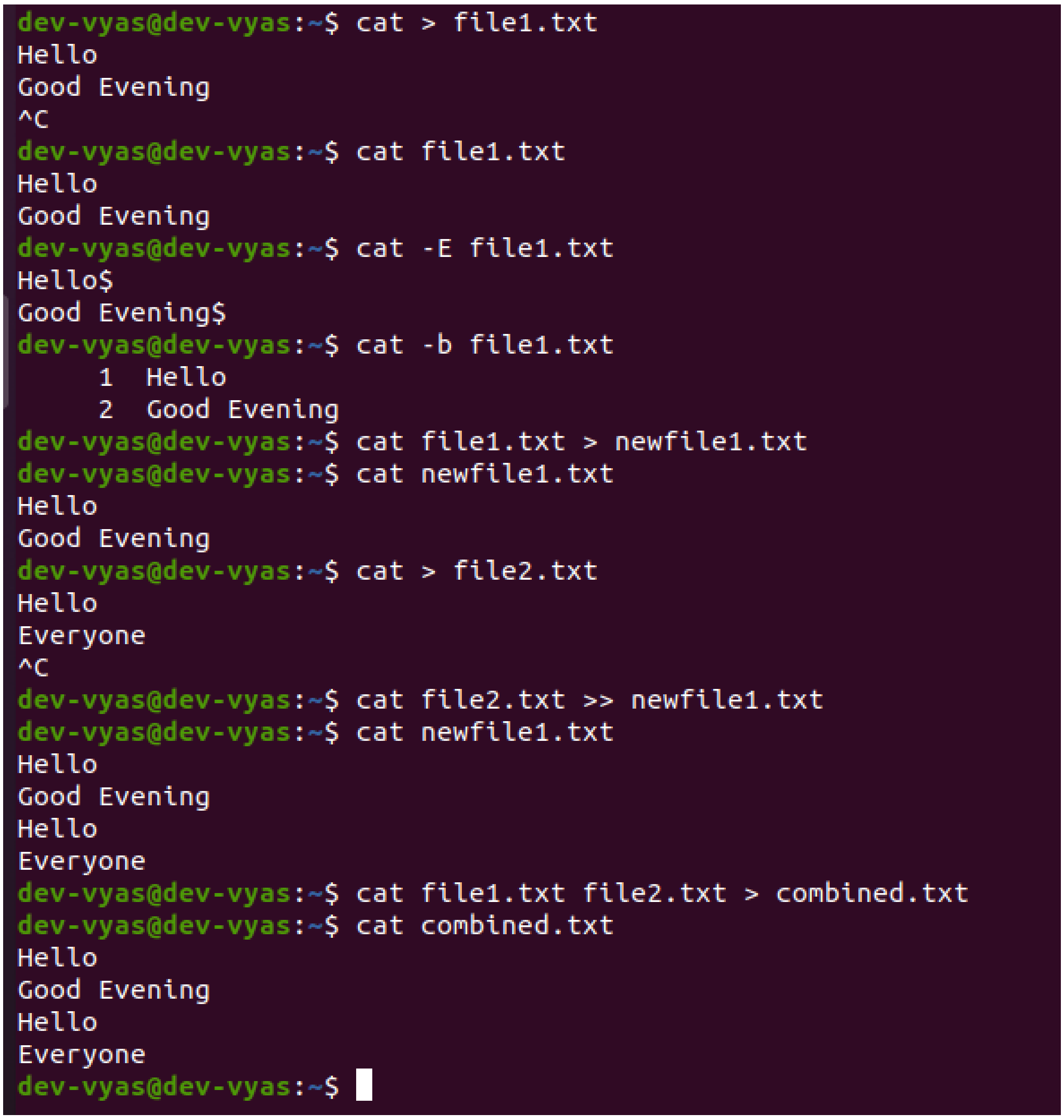
Option	**Use**
date +%a	The abbreviated weekday name (e.g., Sun)
date +%A	The full weekday name (e.g., Sunday)
date +%b	The abbreviated month name (e.g., Jan)
date +%B	Locale's full month name (e.g., January)
date +%C	The current century; like %Y, except omit last two digits (e.g., 20)
date +%w	day of week (0..6); 0 is Sunday
date +%d	Display the day of the month
date +%m	Displays the month of year (01 to 12)
date +%y	Displays last two digits of the year(00 to 99)
date +%Y	Display four-digit year.
date +%T	Display the time in 24 hour format as HH:MM:SS
date +%H	Display the hour
date +%M	Display the minute
date +%S	Display the seconds
date +%V	ISO week number, with Monday as first day of week (01..53)
date +%P	locale's equivalent of either AM or PM



```
dev-vyas@dev-vyas:~$ date
Monday 28 March 2022 06:17:07 PM IST
dev-vyas@dev-vyas:~$ date +%a
Mon
dev-vyas@dev-vyas:~$ date +%A
Monday
dev-vyas@dev-vyas:~$ date +%b
Mar
dev-vyas@dev-vyas:~$ date +%B
March
dev-vyas@dev-vyas:~$ date +%C
20
dev-vyas@dev-vyas:~$ date +%c
Monday 28 March 2022 06:17:26 PM
dev-vyas@dev-vyas:~$ date +%d
28
dev-vyas@dev-vyas:~$ date +%m
03
dev-vyas@dev-vyas:~$ date +%y
22
dev-vyas@dev-vyas:~$ date +%Y
2022
dev-vyas@dev-vyas:~$ date +%T
18:17:55
dev-vyas@dev-vyas:~$ date +%H
18
dev-vyas@dev-vyas:~$ date +%M
18
dev-vyas@dev-vyas:~$ date +%S
44
dev-vyas@dev-vyas:~$ date +%V
13
dev-vyas@dev-vyas:~$ date +%p
PM
dev-vyas@dev-vyas:~$ date +%P
pm
dev-vyas@dev-vyas:~$
```

- **cat command:-** It is used to create, display and concatenate file contents.
  Syntax:- cat [OPTION] [FILE]

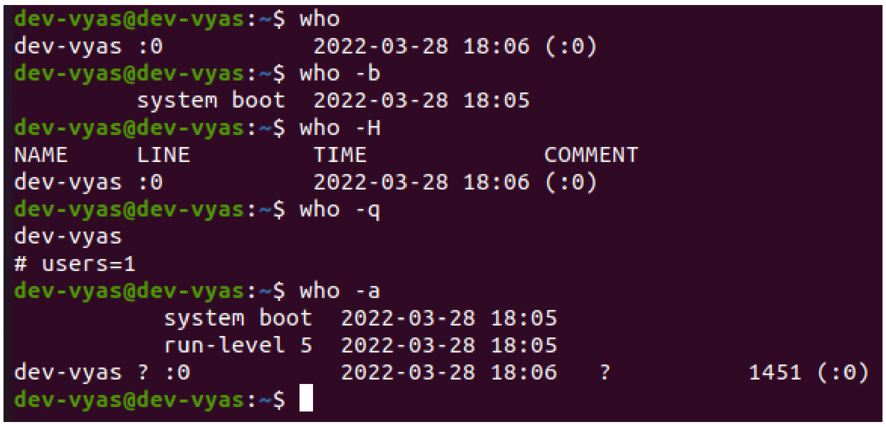
Option	**Use**
cat –b	Omits line numbers for blank space in the output
cat –E	Displays a $ (dollar sign) at the end of each line
cat –n	Line numbers for all the output lines
cat –s	Suppress repeated empty output lines
cat –T	Displays the tab characters as ^I in the output



```
dev-vyas@dev-vyas:~$ cat > file1.txt
Hello
Good Evening
^C
dev-vyas@dev-vyas:~$ cat file1.txt
Hello
Good Evening
dev-vyas@dev-vyas:~$ cat -E file1.txt
Hello$
Good Evening$
dev-vyas@dev-vyas:~$ cat -b file1.txt
     1  Hello
     2  Good Evening
dev-vyas@dev-vyas:~$ cat file1.txt > newfile1.txt
dev-vyas@dev-vyas:~$ cat newfile1.txt
Hello
Good Evening
dev-vyas@dev-vyas:~$ cat > file2.txt
Hello
Everyone
^C
dev-vyas@dev-vyas:~$ cat file2.txt >> newfile1.txt
dev-vyas@dev-vyas:~$ cat newfile1.txt
Hello
Good Evening
Hello
Everyone
dev-vyas@dev-vyas:~$ cat file1.txt file2.txt > combined.txt
dev-vyas@dev-vyas:~$ cat combined.txt
Hello
Good Evening
Hello
Everyone
dev-vyas@dev-vyas:~$ 
```

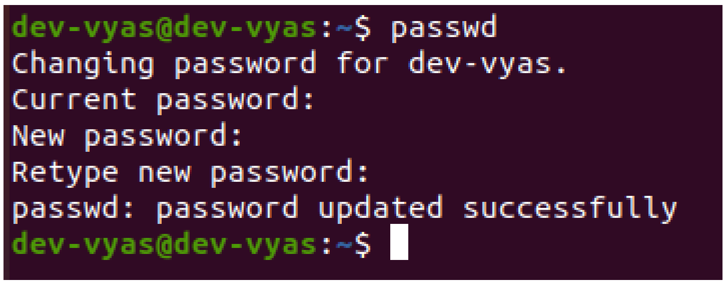
- **who command:- :** It display the users that are currently logged into your Unix computer system.
  Syntax:- who [-options] [filename]

Option	**Use**
who –b	Display the time of the last system boot
who –H	Print a line of column headings
who –q	Displays all login names, and a count of all logged-on users
who –a	Display all details of current logged in user



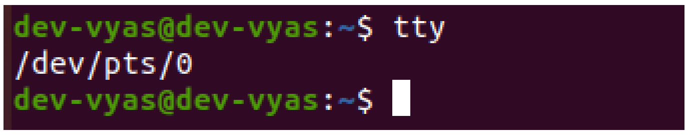
```
dev-vyas@dev-vyas:~$ who
dev-vyas :0           2022-03-28 18:06 (:0)
dev-vyas@dev-vyas:~$ who -b
         system boot  2022-03-28 18:05
dev-vyas@dev-vyas:~$ who -H
NAME     LINE         TIME             COMMENT
dev-vyas :0           2022-03-28 18:06 (:0)
dev-vyas@dev-vyas:~$ who -q
dev-vyas
# users=1
dev-vyas@dev-vyas:~$ who -a
           system boot  2022-03-28 18:05
           run-level 5  2022-03-28 18:05
dev-vyas ? :0           2022-03-28 18:06   ?          1451 (:0)
dev-vyas@dev-vyas:~$
```

- **passwd command:-** The passwd command is used to change the password of a user account.
  Syntax:- passwd [-options] [username]



```
dev-vyas@dev-vyas:~$ passwd
Changing password for dev-vyas.
Current password:
New password:
Retype new password:
passwd: password updated successfully
dev-vyas@dev-vyas:~$
```

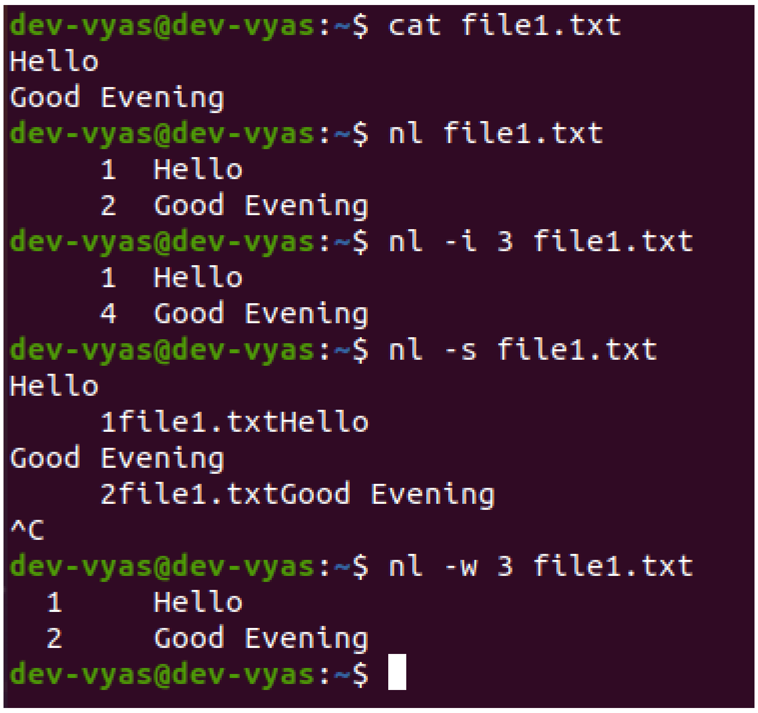
- **tty command:-** Print the file name of the terminal connected to standard input.
  Syntax:- tty



```
dev-vyas@dev-vyas:~$ tty
/dev/pts/0
dev-vyas@dev-vyas:~$
```

- **nl command:- :** nl command numbers the lines in a file.
  Syntax:- nl [OPTION]... [FILE]...

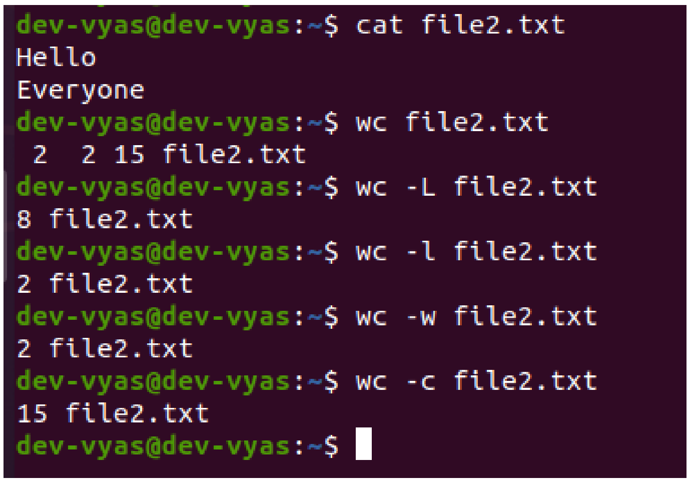
Option	Use
nl -i	Line number increment at each line
nl -s	Add STRING after (possible) line number
nl -w	Use NUMBER columns for line numbers



```
dev-vyas@dev-vyas:~$ cat file1.txt
Hello
Good Evening
dev-vyas@dev-vyas:~$ nl file1.txt
     1  Hello
     2  Good Evening
dev-vyas@dev-vyas:~$ nl -i 3 file1.txt
     1  Hello
     4  Good Evening
dev-vyas@dev-vyas:~$ nl -s file1.txt
Hello
     1file1.txtHello
Good Evening
     2file1.txtGood Evening
^C
dev-vyas@dev-vyas:~$ nl -w 3 file1.txt
  1     Hello
  2     Good Evening
dev-vyas@dev-vyas:~$ 
```

- **wc command:-** It s used to find out number of newline count, word count, byte and characters count in a files specified by the file arguments.
  Syntax:- wc [options] filenames

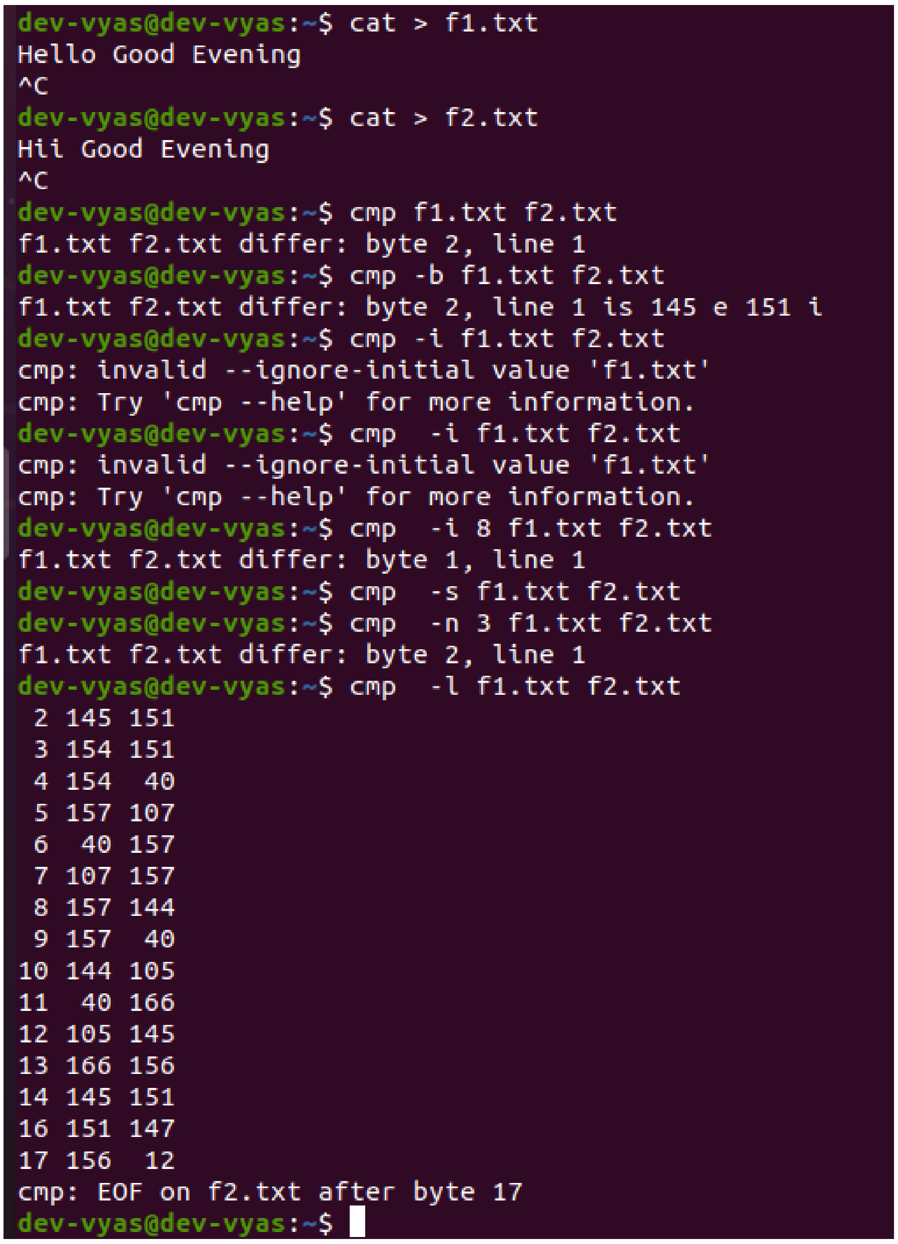
Option	Use
wc -l	Prints the number of lines in a file
wc -w	Prints the number of words in a file
wc -c	Displays the count of bytes in a file
wc -L	Prints only the length of the longest line in a file



```
dev-vyas@dev-vyas:~$ cat file2.txt
Hello
Everyone
dev-vyas@dev-vyas:~$ wc file2.txt
 2  2 15 file2.txt
dev-vyas@dev-vyas:~$ wc -L file2.txt
8 file2.txt
dev-vyas@dev-vyas:~$ wc -l file2.txt
2 file2.txt
dev-vyas@dev-vyas:~$ wc -w file2.txt
2 file2.txt
dev-vyas@dev-vyas:~$ wc -c file2.txt
15 file2.txt
dev-vyas@dev-vyas:~$ 
```

- **cmp command:-** cmp command in Linux/UNIX is used to compare the two files byte by byte and helps you to find out whether the two files are identical or not.
  If a difference is found, it reports the byte and line number where the first difference is found. If no differences are found, by default, cmp returns no output.
  Syntax:- cmp [OPTION]... FILE1 [FILE2 [SKIP1 [SKIP2]]]

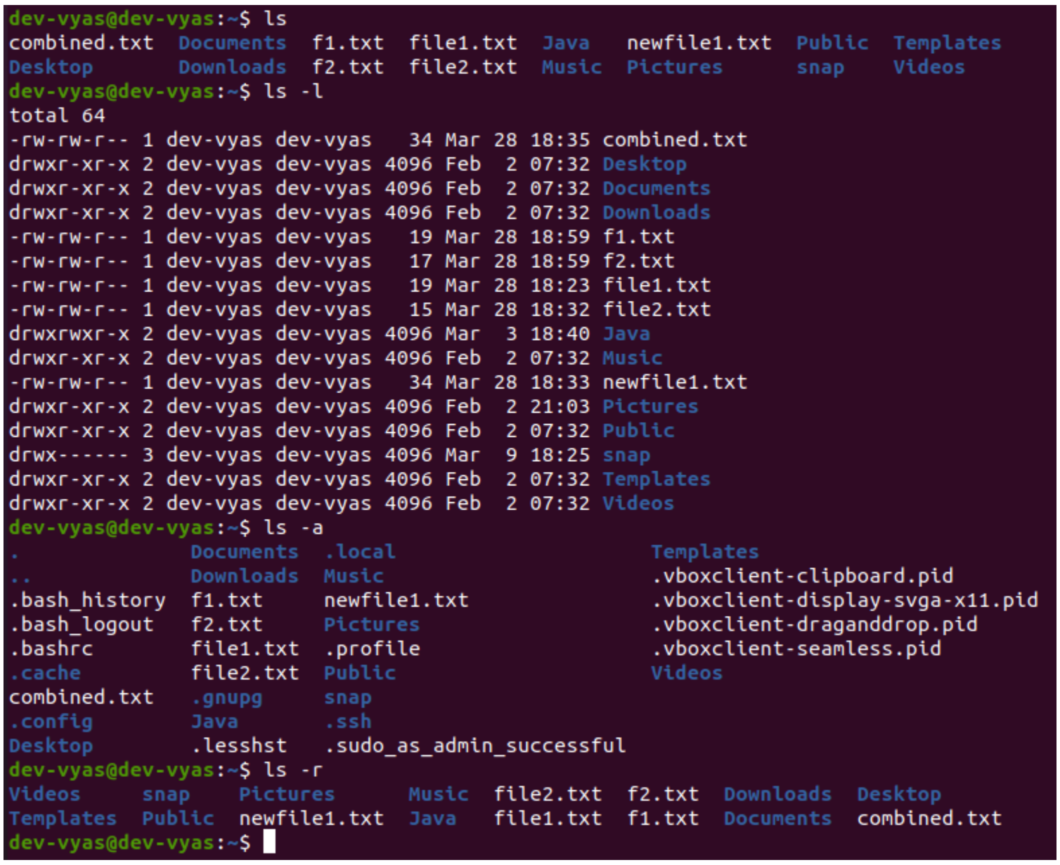
Option	**Use**
cmp -b	Print differing bytes
cmp -i	Skip a particular number of initial bytes from both the files
cmp -s	Do not print anything; only return an exit status indicating whether the files differ
cmp -n	Compare at most LIMIT bytes
cmp -l	Print byte position and byte value for all differing bytes



```
dev-vyas@dev-vyas:~$ cat > f1.txt
Hello Good Evening
^C
dev-vyas@dev-vyas:~$ cat > f2.txt
Hii Good Evening
^C
dev-vyas@dev-vyas:~$ cmp f1.txt f2.txt
f1.txt f2.txt differ: byte 2, line 1
dev-vyas@dev-vyas:~$ cmp -b f1.txt f2.txt
f1.txt f2.txt differ: byte 2, line 1 is 145 e 151 i
dev-vyas@dev-vyas:~$ cmp -i f1.txt f2.txt
cmp: invalid --ignore-initial value 'f1.txt'
cmp: Try 'cmp --help' for more information.
dev-vyas@dev-vyas:~$ cmp  -i f1.txt f2.txt
cmp: invalid --ignore-initial value 'f1.txt'
cmp: Try 'cmp --help' for more information.
dev-vyas@dev-vyas:~$ cmp  -i 8 f1.txt f2.txt
f1.txt f2.txt differ: byte 1, line 1
dev-vyas@dev-vyas:~$ cmp  -s f1.txt f2.txt
dev-vyas@dev-vyas:~$ cmp  -n 3 f1.txt f2.txt
f1.txt f2.txt differ: byte 2, line 1
dev-vyas@dev-vyas:~$ cmp  -l f1.txt f2.txt
 2 145 151
 3 154 151
 4 154  40
 5 157 107
 6  40 157
 7 107 157
 8 157 144
 9 157  40
10 144 105
11  40 166
12 105 145
13 166 156
14 145 151
16 151 147
17 156  12
cmp: EOF on f2.txt after byte 17
dev-vyas@dev-vyas:~$ 
```

- **ls command:-** List directory contents.
  Syntax:- ls [Options] [file|dir]

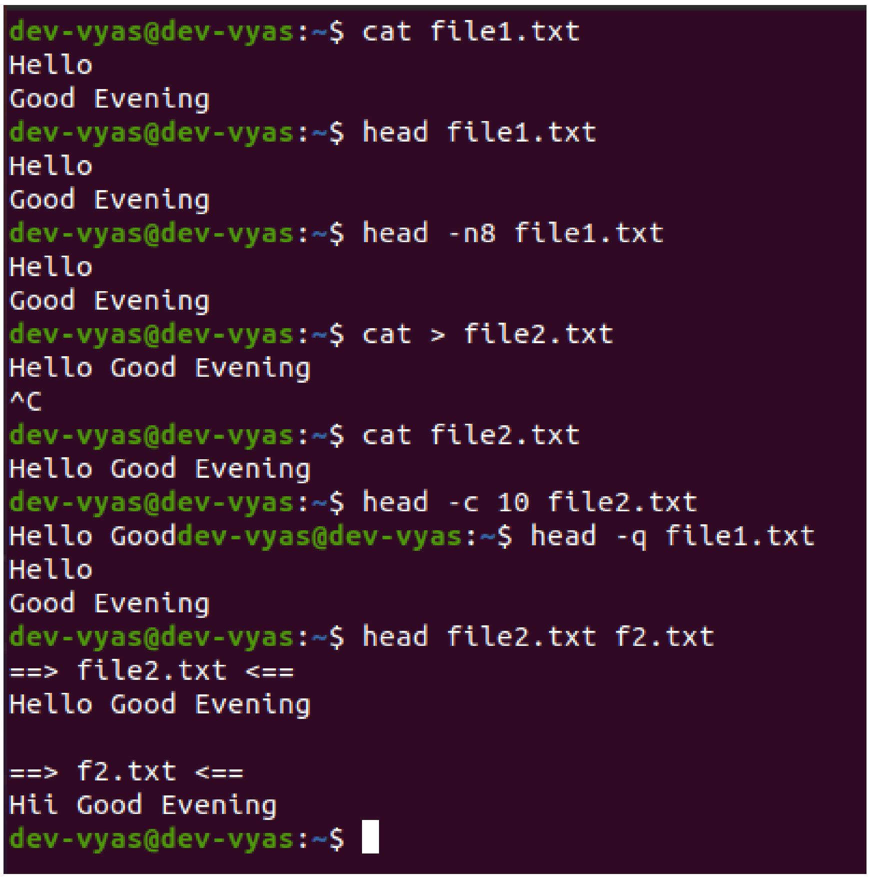
Option	**Use**
ls -l	To show long listing information about the file/directory
ls -a	List all files including hidden file starting with '.'
ls -r	List in reverse order
ls -t	Sort by time & date
ls -s	Sort by file size



```
dev-vyas@dev-vyas:~$ ls
combined.txt  Documents  f1.txt  file1.txt  Java   newfile1.txt  Public  Templates
Desktop       Downloads  f2.txt  file2.txt  Music  Pictures      snap    Videos
dev-vyas@dev-vyas:~$ ls -l
total 64
-rw-rw-r-- 1 dev-vyas dev-vyas   34 Mar 28 18:35 combined.txt
drwxr-xr-x 2 dev-vyas dev-vyas 4096 Feb  2 07:32 Desktop
drwxr-xr-x 2 dev-vyas dev-vyas 4096 Feb  2 07:32 Documents
drwxr-xr-x 2 dev-vyas dev-vyas 4096 Feb  2 07:32 Downloads
-rw-rw-r-- 1 dev-vyas dev-vyas   19 Mar 28 18:59 f1.txt
-rw-rw-r-- 1 dev-vyas dev-vyas   17 Mar 28 18:59 f2.txt
-rw-rw-r-- 1 dev-vyas dev-vyas   19 Mar 28 18:23 file1.txt
-rw-rw-r-- 1 dev-vyas dev-vyas   15 Mar 28 18:32 file2.txt
drwxrwxr-x 2 dev-vyas dev-vyas 4096 Mar  3 18:40 Java
drwxr-xr-x 2 dev-vyas dev-vyas 4096 Feb  2 07:32 Music
-rw-rw-r-- 1 dev-vyas dev-vyas   34 Mar 28 18:33 newfile1.txt
drwxr-xr-x 2 dev-vyas dev-vyas 4096 Feb  2 21:03 Pictures
drwxr-xr-x 2 dev-vyas dev-vyas 4096 Feb  2 07:32 Public
drwx------ 3 dev-vyas dev-vyas 4096 Mar  9 18:25 snap
drwxr-xr-x 2 dev-vyas dev-vyas 4096 Feb  2 07:32 Templates
drwxr-xr-x 2 dev-vyas dev-vyas 4096 Feb  2 07:32 Videos
dev-vyas@dev-vyas:~$ ls -a
.              Documents  .local                     Templates
..             Downloads  Music                      .vboxclient-clipboard.pid
.bash_history  f1.txt     newfile1.txt               .vboxclient-display-svga-x11.pid
.bash_logout   f2.txt     Pictures                   .vboxclient-draganddrop.pid
.bashrc        file1.txt  .profile                   .vboxclient-seamless.pid
.cache         file2.txt  Public                     Videos
combined.txt   .gnupg     snap
.config        Java       .ssh
Desktop        .lesshst   .sudo_as_admin_successful
dev-vyas@dev-vyas:~$ ls -r
Videos     snap    Pictures      Music  file2.txt  f2.txt  Downloads  Desktop
Templates  Public  newfile1.txt  Java   file1.txt  f1.txt  Documents  combined.txt
dev-vyas@dev-vyas:~$ █
```

- **head command:-** head makes it easy to output the first part (10 lines by default) of files.
  Syntax:- head [OPTION]... [FILE]...

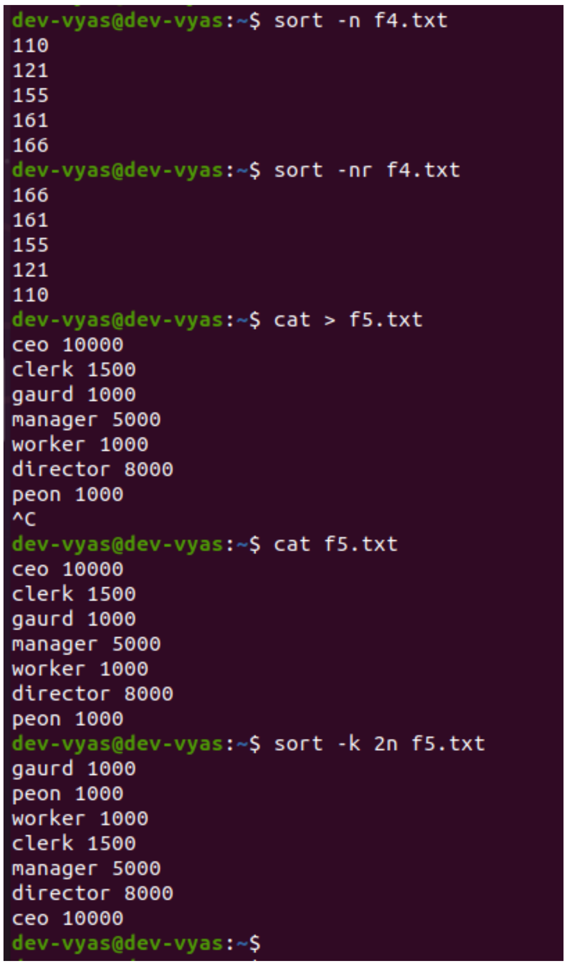
Option	**Use**
head -n	Print the first n lines instead of the first 10; with the leading '-', print all but the last n lines of each file
head -c	Print the first n bytes of each file; with a leading '-', print all but the last n bytes of each file
head -q	Never print headers identifying file names



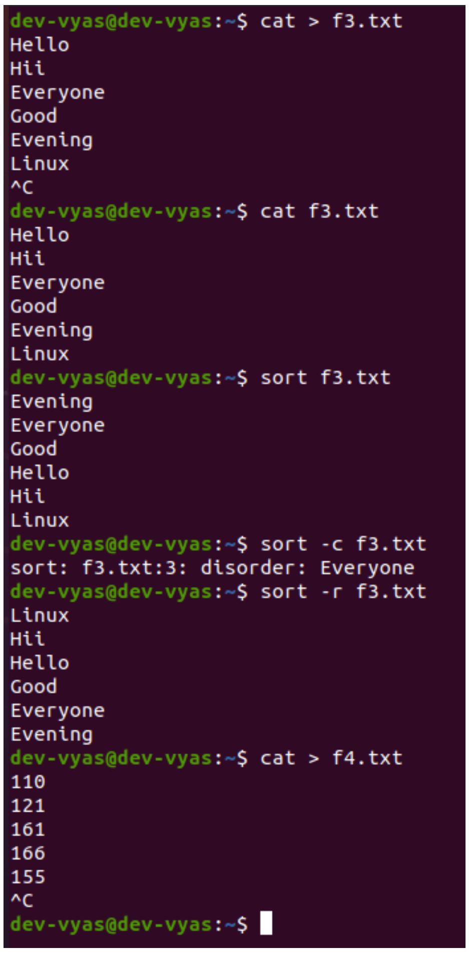
```
dev-vyas@dev-vyas:~$ cat file1.txt
Hello
Good Evening
dev-vyas@dev-vyas:~$ head file1.txt
Hello
Good Evening
dev-vyas@dev-vyas:~$ head -n8 file1.txt
Hello
Good Evening
dev-vyas@dev-vyas:~$ cat > file2.txt
Hello Good Evening
^C
dev-vyas@dev-vyas:~$ cat file2.txt
Hello Good Evening
dev-vyas@dev-vyas:~$ head -c 10 file2.txt
Hello Gooddev-vyas@dev-vyas:~$ head -q file1.txt
Hello
Good Evening
dev-vyas@dev-vyas:~$ head file2.txt f2.txt
==> file2.txt <==
Hello Good Evening

==> f2.txt <==
Hii Good Evening
dev-vyas@dev-vyas:~$
```

- **sort command:-** Sort command is used to sort a file, arranging the records in a particular order.By default, the sort command sorts file assuming the contents are ASCII. Using options in sort command, it can also be used to sort numerically.
  Syntax:- sort [OPTION]... [FILE]...

Option	**Use**
sort -c	To check if the file given is already sorted or not
sort –r	Reverse the result of comparisons
sort -n	Compare according to string numerical value
sort -nr	To sort a file with numeric data in reverse order
sort -k	Sorting a table on the basis of any column
sort -b	Ignore leading blanks



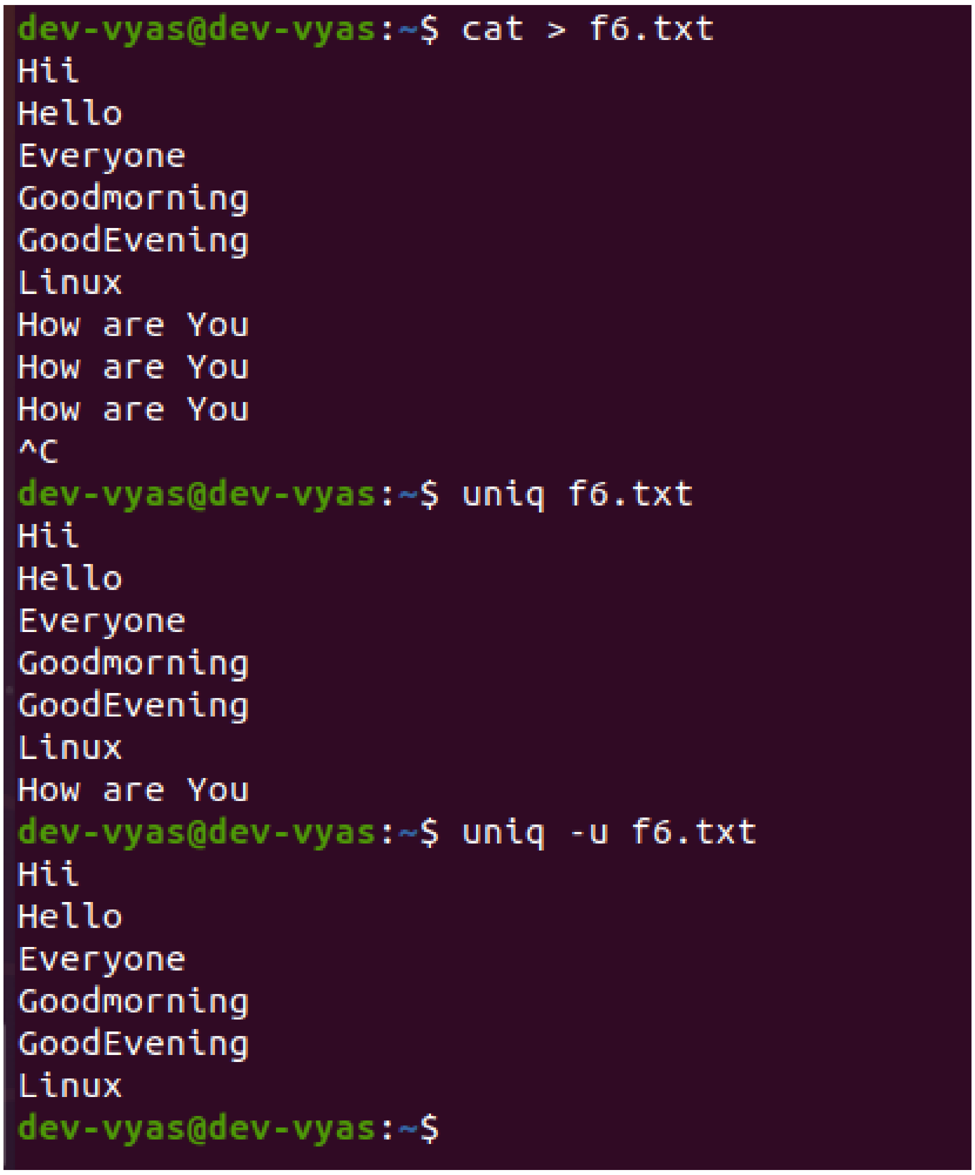
```
dev-vyas@dev-vyas:~$ sort -n f4.txt
110
121
155
161
166
dev-vyas@dev-vyas:~$ sort -nr f4.txt
166
161
155
121
110
dev-vyas@dev-vyas:~$ cat > f5.txt
ceo 10000
clerk 1500
gaurd 1000
manager 5000
worker 1000
director 8000
peon 1000
^C
dev-vyas@dev-vyas:~$ cat f5.txt
ceo 10000
clerk 1500
gaurd 1000
manager 5000
worker 1000
director 8000
peon 1000
dev-vyas@dev-vyas:~$ sort -k 2n f5.txt
gaurd 1000
peon 1000
worker 1000
clerk 1500
manager 5000
director 8000
ceo 10000
dev-vyas@dev-vyas:~$
```



```
dev-vyas@dev-vyas:~$ cat > f3.txt
Hello
Hii
Everyone
Good
Evening
Linux
^C
dev-vyas@dev-vyas:~$ cat f3.txt
Hello
Hii
Everyone
Good
Evening
Linux
dev-vyas@dev-vyas:~$ sort f3.txt
Evening
Everyone
Good
Hello
Hii
Linux
dev-vyas@dev-vyas:~$ sort -c f3.txt
sort: f3.txt:3: disorder: Everyone
dev-vyas@dev-vyas:~$ sort -r f3.txt
Linux
Hii
Hello
Good
Everyone
Evening
dev-vyas@dev-vyas:~$ cat > f4.txt
110
121
161
166
155
^C
dev-vyas@dev-vyas:~$
```

- **uniq command:-** uniq reports or filters out repeated lines in a file. It can remove duplicates, show a count of occurrences, show only repeated lines, ignore certain characters and compare on specific fields.
  Syntax:- uniq [OPTION]... [INPUT [OUTPUT]]

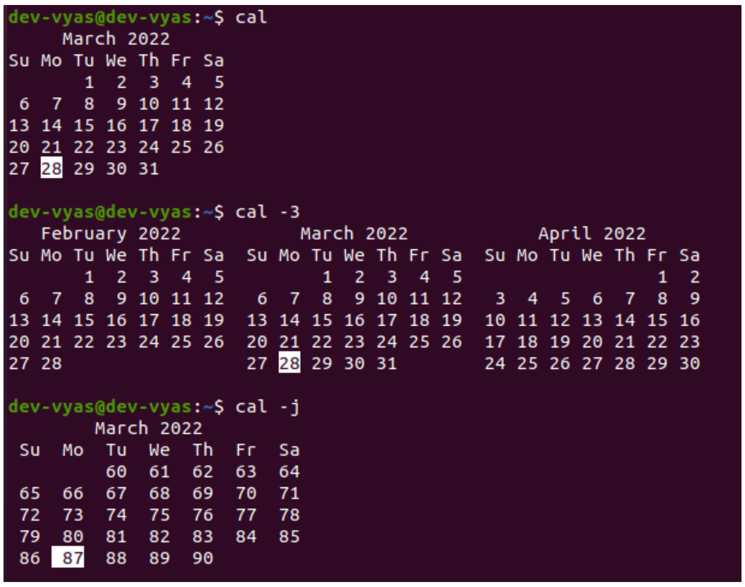
Option	**Use**
uniq -u	Prints only unique lines
uniq –d	Only print duplicated lines
uniq -D	Print all duplicate lines
uniq -c	Prefix lines with a number representing how many times they occurred
uniq -i	Ignore case when comparing



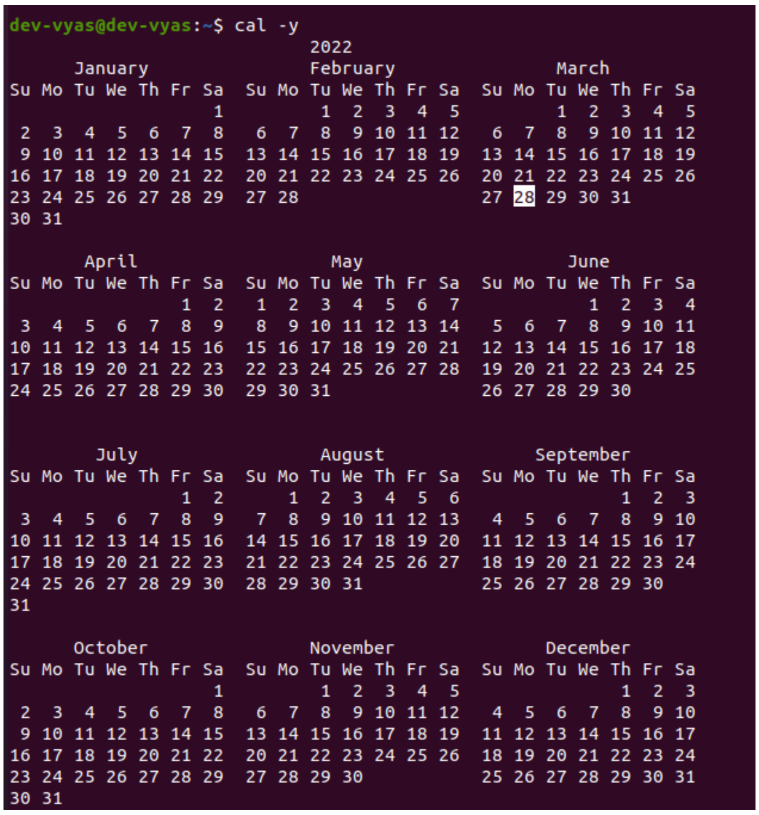
```
dev-vyas@dev-vyas:~$ cat > f6.txt
Hii
Hello
Everyone
Goodmorning
GoodEvening
Linux
How are You
How are You
How are You
^C
dev-vyas@dev-vyas:~$ uniq f6.txt
Hii
Hello
Everyone
Goodmorning
GoodEvening
Linux
How are You
dev-vyas@dev-vyas:~$ uniq -u f6.txt
Hii
Hello
Everyone
Goodmorning
GoodEvening
Linux
dev-vyas@dev-vyas:~$
```

- **cal command:-** Displays a simple, formatted calendar in your terminal.
  Syntax:- cal [options] [[[day] month] year]

Option	**Use**
Cal -1	Display single month output. (This is the default.)
Cal -3	Display three months spanning the date.
Cal –s	Display Sunday as the first day of the week.
Cal –m	Display Monday as the first day of the week.
Cal –j	Use day-of-year numbering for all calendars. These are also called ordinal days. Ordinal days range from 1 to 366.
Cal -y	Display a calendar for the whole year

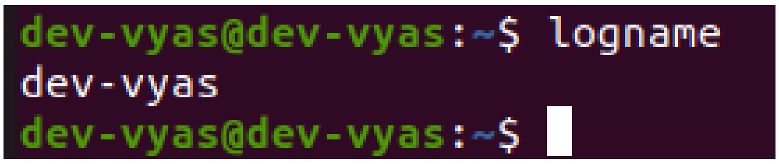


```
dev-vyas@dev-vyas:~$ cal
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

dev-vyas@dev-vyas:~$ cal -3
   February 2022          March 2022          April 2022
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  3  4  5                  1  2
 6  7  8  9 10 11 12   6  7  8  9 10 11 12   3  4  5  6  7  8  9
13 14 15 16 17 18 19  13 14 15 16 17 18 19  10 11 12 13 14 15 16
20 21 22 23 24 25 26  20 21 22 23 24 25 26  17 18 19 20 21 22 23
27 28                 27 28 29 30 31        24 25 26 27 28 29 30

dev-vyas@dev-vyas:~$ cal -j
       March 2022
 Su  Mo  Tu  We  Th  Fr  Sa
         60  61  62  63  64
 65  66  67  68  69  70  71
 72  73  74  75  76  77  78
 79  80  81  82  83  84  85
 86  87  88  89  90
```



```
dev-vyas@dev-vyas:~$ cal -y
                            2022
      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         1  2  3  4  5         1  2  3  4  5
 2  3  4  5  6  7  8   6  7  8  9 10 11 12   6  7  8  9 10 11 12
 9 10 11 12 13 14 15  13 14 15 16 17 18 19  13 14 15 16 17 18 19
16 17 18 19 20 21 22  20 21 22 23 24 25 26  20 21 22 23 24 25 26
23 24 25 26 27 28 29  27 28                 27 28 29 30 31
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
```

- **logname command:-** It tells the name of user who is logged in at that time.
  Syntax:- logname



```
dev-vyas@dev-vyas:~$ logname
dev-vyas
dev-vyas@dev-vyas:~$
```

Practical 2

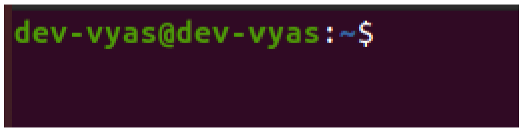
AIM: Study of Advance commands and filters of Linux/UNIX.

1. clear Command:

- Clear the terminal screen.
- **Syntax:**

  Clear

- Example:

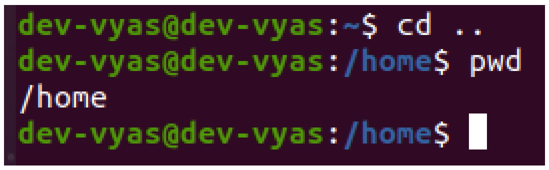


```
dev-vyas@dev-vyas:~$
```

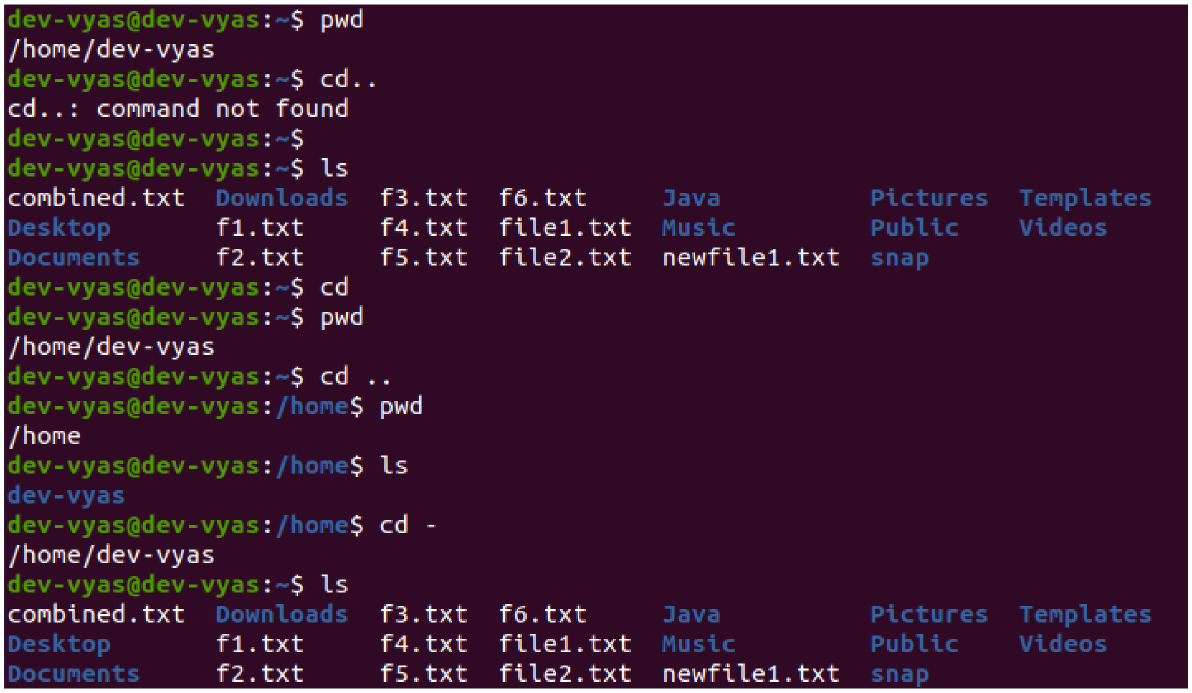
2. cd Command:

- The cd command is used to change the current directory (i.e., the directory in which the user is currently working)
- **Syntax: cd [-Options] [Directory]**
- Example:

Option	Use
cd..	Change Current directory to parent directory
cd ~	Move to users home directory from anywhere
cd lab_1	Change from current working directory to lab_1
cd ../downloads	If we are currently in /home/username/documents then we would be placed in /home/username/downloads.



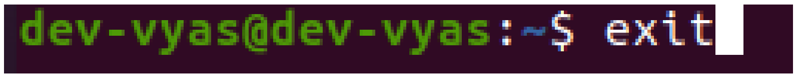
```
dev-vyas@dev-vyas:~$ cd ..
dev-vyas@dev-vyas:/home$ pwd
/home
dev-vyas@dev-vyas:/home$ 
```



```
dev-vyas@dev-vyas:~$ pwd
/home/dev-vyas
dev-vyas@dev-vyas:~$ cd..
cd..: command not found
dev-vyas@dev-vyas:~$
dev-vyas@dev-vyas:~$ ls
combined.txt  Downloads  f3.txt  f6.txt     Java         Pictures  Templates
Desktop       f1.txt     f4.txt  file1.txt  Music        Public    Videos
Documents     f2.txt     f5.txt  file2.txt  newfile1.txt snap
dev-vyas@dev-vyas:~$ cd
dev-vyas@dev-vyas:~$ pwd
/home/dev-vyas
dev-vyas@dev-vyas:~$ cd ..
dev-vyas@dev-vyas:/home$ pwd
/home
dev-vyas@dev-vyas:/home$ ls
dev-vyas
dev-vyas@dev-vyas:/home$ cd -
/home/dev-vyas
dev-vyas@dev-vyas:~$ ls
combined.txt  Downloads  f3.txt  f6.txt     Java         Pictures  Templates
Desktop       f1.txt     f4.txt  file1.txt  Music        Public    Videos
Documents     f2.txt     f5.txt  file2.txt  newfile1.txt snap
```

3. exit Command:

- It is used to terminate the program, shell or log you out of a network normally.
- **Syntax: exit**
- Example:

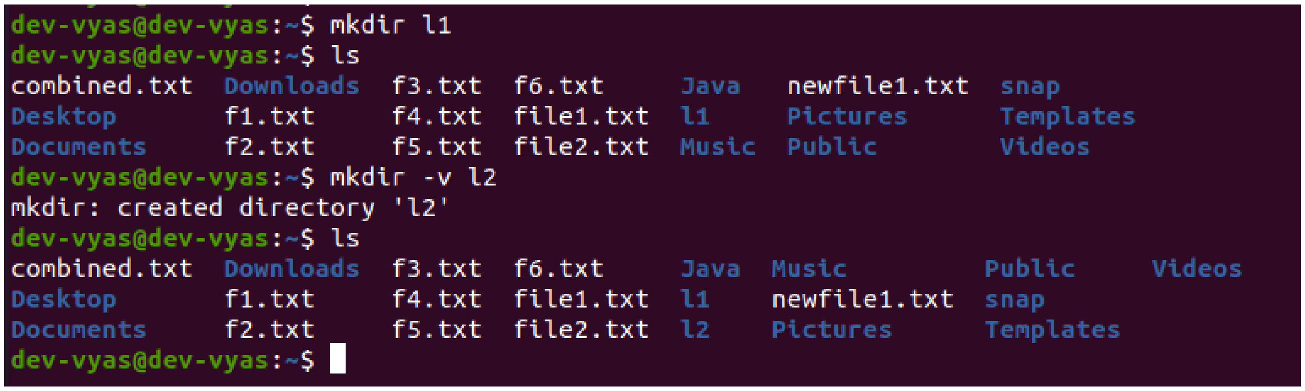


```
dev-vyas@dev-vyas:~$ exit
```

4. mkdir Command:

- This command is used to make Directories.
- **Syntax: mkdir [-OPTION] DIRECTORY**
- Example:

Option	Use
mkdir -v	Print a message for each created directory
mkdir -p	No error if existing, make parent directories as needed
mkdir -m	To control the permissions of new directories



```
dev-vyas@dev-vyas:~$ mkdir l1
dev-vyas@dev-vyas:~$ ls
combined.txt  Downloads  f3.txt  f6.txt     Java   newfile1.txt  snap
Desktop       f1.txt     f4.txt  file1.txt  l1     Pictures      Templates
Documents     f2.txt     f5.txt  file2.txt  Music  Public        Videos
dev-vyas@dev-vyas:~$ mkdir -v l2
mkdir: created directory 'l2'
dev-vyas@dev-vyas:~$ ls
combined.txt  Downloads  f3.txt  f6.txt     Java  Music         Public     Videos
Desktop       f1.txt     f4.txt  file1.txt  l1    newfile1.txt  snap
Documents     f2.txt     f5.txt  file2.txt  l2    Pictures      Templates
dev-vyas@dev-vyas:~$ 
```

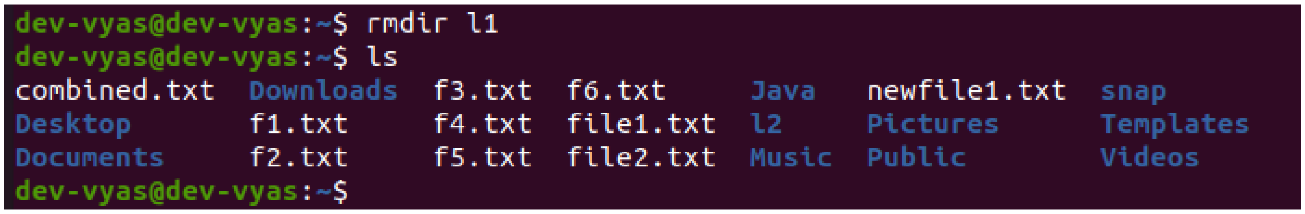
5. rmdir Command:

- This command removes empty directories from your filesystem.
- **Syntax:**

  rmdir [-OPTION] DIRECTORY •

Example:

Option	**Use**
rmdir -p	Remove directory and its ancestors… e.g., ‘rmdir -p a/b/c’ is similar to ‘rmdir a/b/c a/b a’



```
dev-vyas@dev-vyas:~$ rmdir l1
dev-vyas@dev-vyas:~$ ls
combined.txt  Downloads  f3.txt  f6.txt     Java   newfile1.txt  snap
Desktop       f1.txt     f4.txt  file1.txt  l2     Pictures      Templates
Documents     f2.txt     f5.txt  file2.txt  Music  Public        Videos
dev-vyas@dev-vyas:~$
```

6. bc Command:

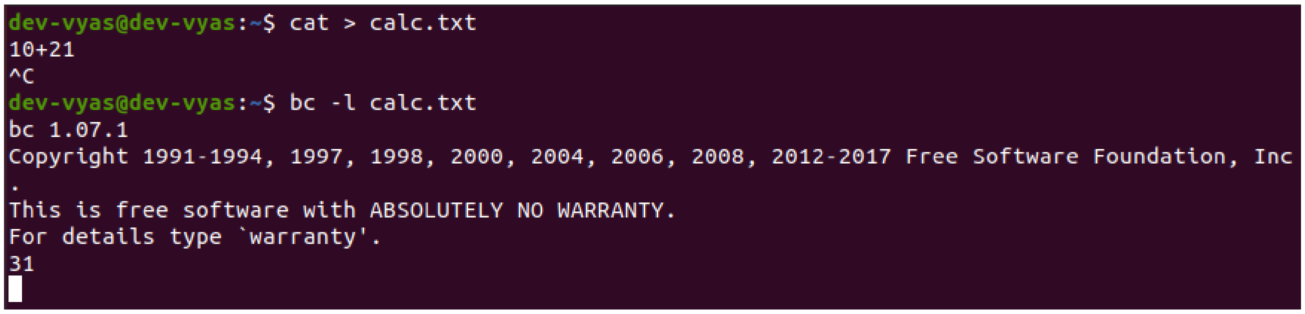
- bc command is used for command line calculator. It is similar to basic calculator. By using which we can do basic mathematical calculations.
- **Syntax:**

  bc [options]

- Example:

Option	**Use**
-q	To avoid bc welcome message

-l	To include math library functionalities

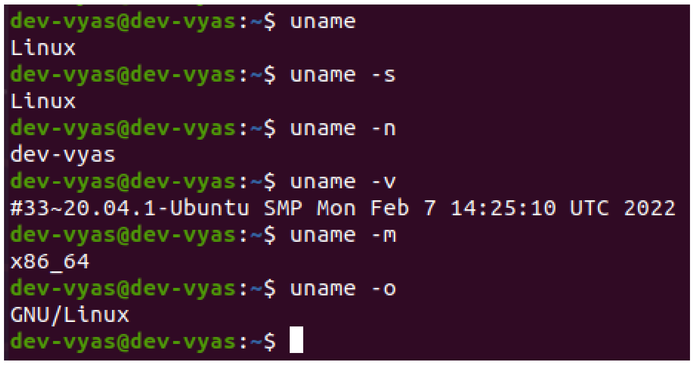


```
dev-vyas@dev-vyas:~$ cat > calc.txt
10+21
^C
dev-vyas@dev-vyas:~$ bc -l calc.txt
bc 1.07.1
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2012-2017 Free Software Foundation, Inc
.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
31
```

7. uname Command:

- Print information about the current system.
- **Syntax: uname [-OPTION]**
- Example:

Option	Use
uname -s	Print the kernel name
uname –n	Print the network node hostname
uname -v	Print the kernel version
uname -m	Print the machine hardware name
uname –o	Print the operating system

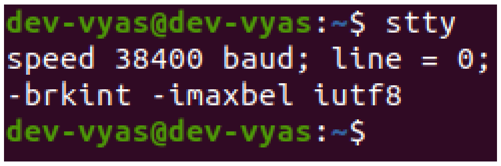


```
dev-vyas@dev-vyas:~$ uname
Linux
dev-vyas@dev-vyas:~$ uname -s
Linux
dev-vyas@dev-vyas:~$ uname -n
dev-vyas
dev-vyas@dev-vyas:~$ uname -v
#33~20.04.1-Ubuntu SMP Mon Feb 7 14:25:10 UTC 2022
dev-vyas@dev-vyas:~$ uname -m
x86_64
dev-vyas@dev-vyas:~$ uname -o
GNU/Linux
dev-vyas@dev-vyas:~$
```

8. sty Command:

- Change and print terminal line settings.
- **Syntax: Sty**

- Example:

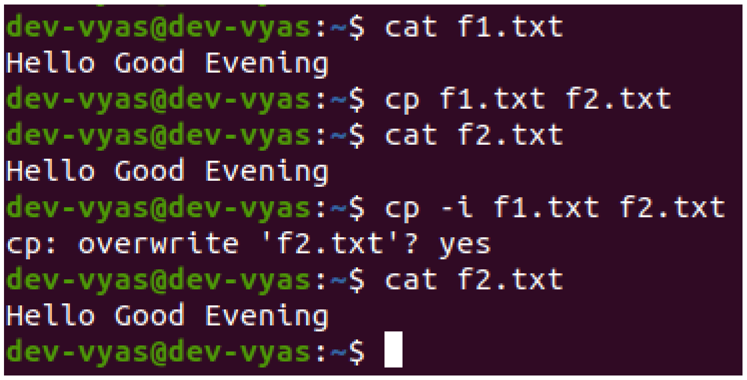


```
dev-vyas@dev-vyas:~$ stty
speed 38400 baud; line = 0;
-brkint -imaxbel iutf8
dev-vyas@dev-vyas:~$
```

9. cp Command:

- (Copy Command) This command is used to copy files and directories.
- **Syntax:**
  cp [option] source destination/directory
- Example:

Option	Use
cp –i	Interactive - ask before overwrite
cp –f	Force copy by removing the destination file if needed
cp –n	Do not overwrite an existing file
cp –u	Update - copy when source is newer than destination
cp –s	Make symbolic links instead of copying
cp –R	Copy directories recursively
cp –v	Print informative messages



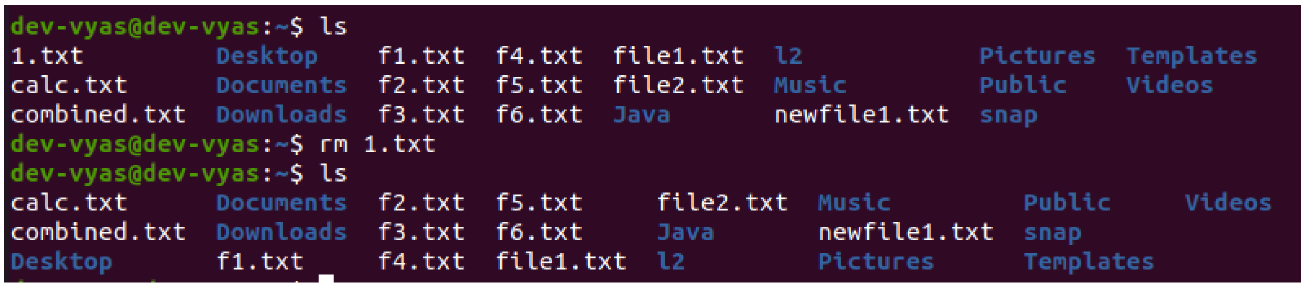
```
dev-vyas@dev-vyas:~$ cat f1.txt
Hello Good Evening
dev-vyas@dev-vyas:~$ cp f1.txt f2.txt
dev-vyas@dev-vyas:~$ cat f2.txt
Hello Good Evening
dev-vyas@dev-vyas:~$ cp -i f1.txt f2.txt
cp: overwrite 'f2.txt'? yes
dev-vyas@dev-vyas:~$ cat f2.txt
Hello Good Evening
dev-vyas@dev-vyas:~$
```

10. rm Command:

- (Remove Command) The ‘rm’ command is used to delete files and directories.
- **Syntax:**
  rm [-OPTION] Filename •

Example:

Option	Use
rm -i	Prompt before every removal
rm -d	Delete a empty directory
rm -r	Remove directories and their contents recursively
rm -f	To remove the file forcefully



```
dev-vyas@dev-vyas:~$ ls
1.txt         Desktop    f1.txt  f4.txt  file1.txt  l2            Pictures  Templates
calc.txt      Documents  f2.txt  f5.txt  file2.txt  Music         Public    Videos
combined.txt  Downloads  f3.txt  f6.txt  Java       newfile1.txt  snap
dev-vyas@dev-vyas:~$ rm 1.txt
dev-vyas@dev-vyas:~$ ls
calc.txt      Documents  f2.txt  f5.txt     file2.txt  Music         Public     Videos
combined.txt  Downloads  f3.txt  f6.txt     Java       newfile1.txt  snap
Desktop       f1.txt     f4.txt  file1.txt  l2         Pictures      Templates
```

11. mv Command:

- (Move Command) mv command is used to move files and directories.
- **Syntax:**
  mv [-options] source dest
- Example:

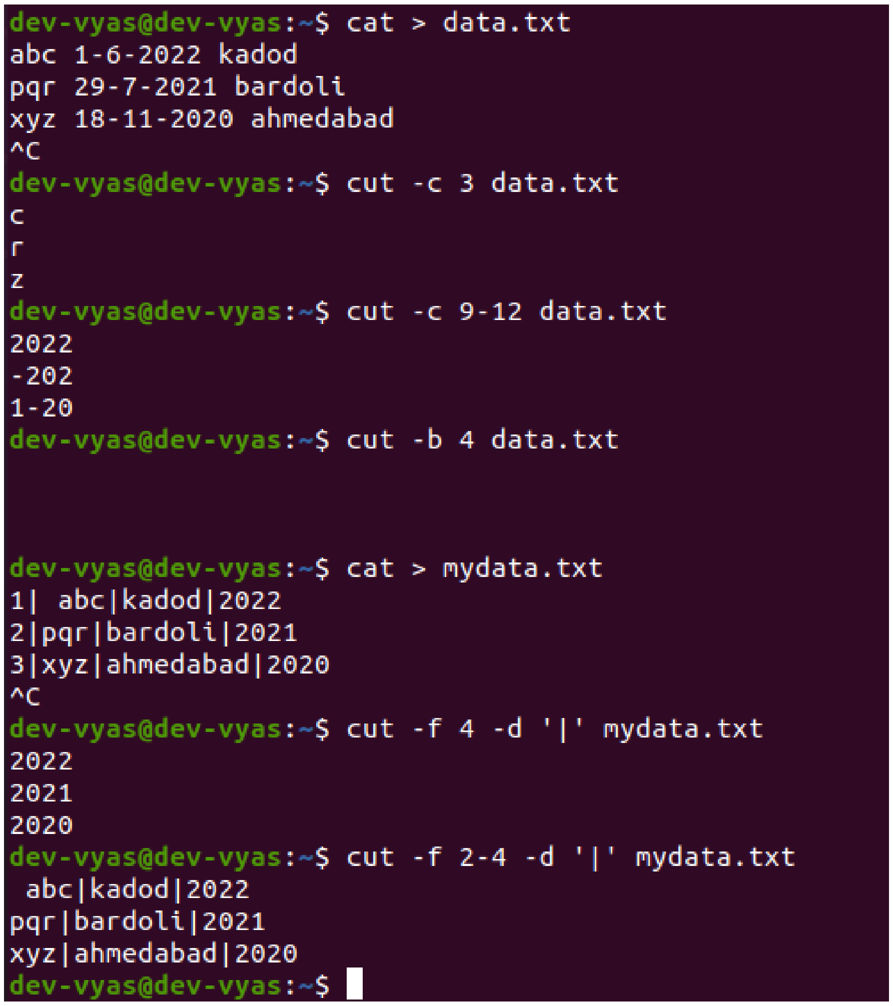
Option	Use
mv -i	Interactive prompt before overwrite
mv -f	Force move by overwriting destination file without prompt

mv -n	Never overwrite any existing file
mv -u	Update - move when source is newer than destination
mv -v	Print informative messages

12. cut Command:

- The cut command extracts a given number of characters or columns from a file.
- **Syntax:**
  cut [-options] [file]
- Example:

Option	Use
cut -c	Select only the characters from each line as specified in LIST
cut -b	Select only the bytes from each line as specified in LIST
cut -f	Cuts the input file using list of field. The default field to be used TAB. The default behavior can be overwritten by use of -d option
cut -d	Specifies a delimiter to by used as a field. Default field is TAB and this option overwrites this default behavior

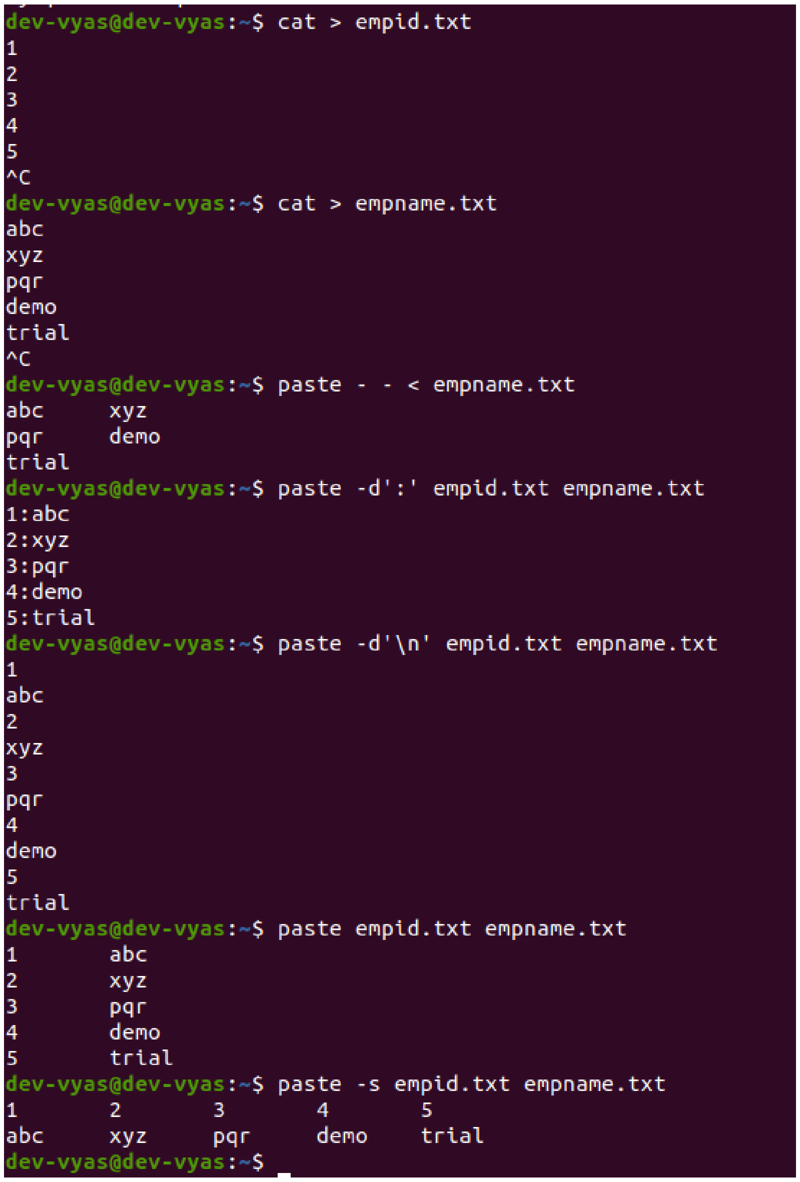


```
dev-vyas@dev-vyas:~$ cat > data.txt
abc 1-6-2022 kadod
pqr 29-7-2021 bardoli
xyz 18-11-2020 ahmedabad
^C
dev-vyas@dev-vyas:~$ cut -c 3 data.txt
c
r
z
dev-vyas@dev-vyas:~$ cut -c 9-12 data.txt
2022
-202
1-20
dev-vyas@dev-vyas:~$ cut -b 4 data.txt



dev-vyas@dev-vyas:~$ cat > mydata.txt
1| abc|kadod|2022
2|pqr|bardoli|2021
3|xyz|ahmedabad|2020
^C
dev-vyas@dev-vyas:~$ cut -f 4 -d '|' mydata.txt
2022
2021
2020
dev-vyas@dev-vyas:~$ cut -f 2-4 -d '|' mydata.txt
 abc|kadod|2022
pqr|bardoli|2021
xyz|ahmedabad|2020
dev-vyas@dev-vyas:~$
```

13. paste Command:

- The paste command displays the corresponding lines of multiple files sideby-side.
- **Syntax:**
  paste [-options] [file]
- Example:

Option	Use
paste -d	Reuse characters from LIST instead of tabs
paste -s	Paste one file at a time instead of in parallel

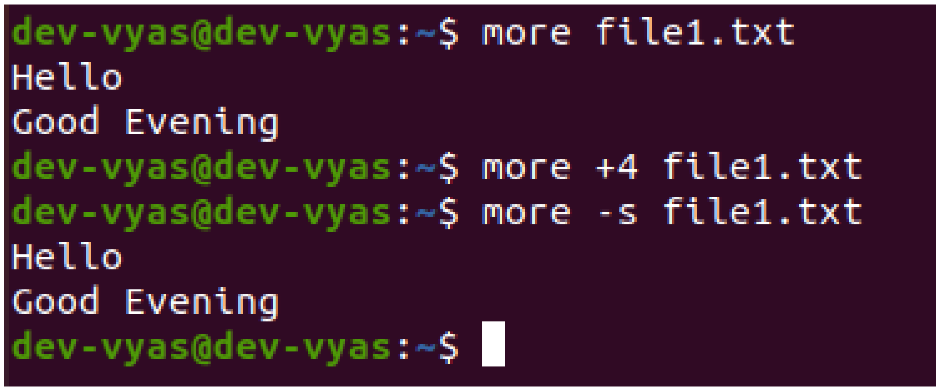


```
dev-vyas@dev-vyas:~$ cat > empid.txt
1
2
3
4
5
^C
dev-vyas@dev-vyas:~$ cat > empname.txt
abc
xyz
pqr
demo
trial
^C
dev-vyas@dev-vyas:~$ paste - - < empname.txt
abc     xyz
pqr     demo
trial
dev-vyas@dev-vyas:~$ paste -d':' empid.txt empname.txt
1:abc
2:xyz
3:pqr
4:demo
5:trial
dev-vyas@dev-vyas:~$ paste -d'\n' empid.txt empname.txt
1
abc
2
xyz
3
pqr
4
demo
5
trial
dev-vyas@dev-vyas:~$ paste empid.txt empname.txt
1       abc
2       xyz
3       pqr
4       demo
5       trial
dev-vyas@dev-vyas:~$ paste -s empid.txt empname.txt
1       2       3       4       5
abc     xyz     pqr     demo    trial
dev-vyas@dev-vyas:~$
```

14. more Command:

- The more command is a command line utility for viewing the contents of a file or files once screen at a time.
- **Syntax:**
  more [-options] [file]
- Example:

Option	**Use**
more –c	Clear screen before displaying
more –number	To Specify how many lines are printed in the screen for a given file
more –s	Doesn’t display extra blank lines



```
dev-vyas@dev-vyas:~$ more file1.txt
Hello
Good Evening
dev-vyas@dev-vyas:~$ more +4 file1.txt
dev-vyas@dev-vyas:~$ more -s file1.txt
Hello
Good Evening
dev-vyas@dev-vyas:~$
```

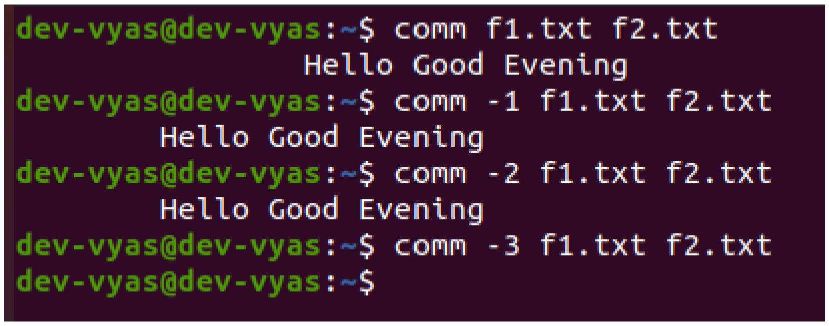
15. comm Command:

- Compare two sorted files line by line.
- **Syntax:**

  comm [OPTION]... FILE1 FILE2

- Example:

Option	Use
comm -1	Suppress column 1 (lines unique to FILE1)
comm -2	Suppress column 2 (lines unique to FILE2)
comm -3	Suppress column 3 (lines that appear in both files)



```
dev-vyas@dev-vyas:~$ comm f1.txt f2.txt
                Hello Good Evening
dev-vyas@dev-vyas:~$ comm -1 f1.txt f2.txt
        Hello Good Evening
dev-vyas@dev-vyas:~$ comm -2 f1.txt f2.txt
        Hello Good Evening
dev-vyas@dev-vyas:~$ comm -3 f1.txt f2.txt
dev-vyas@dev-vyas:~$
```

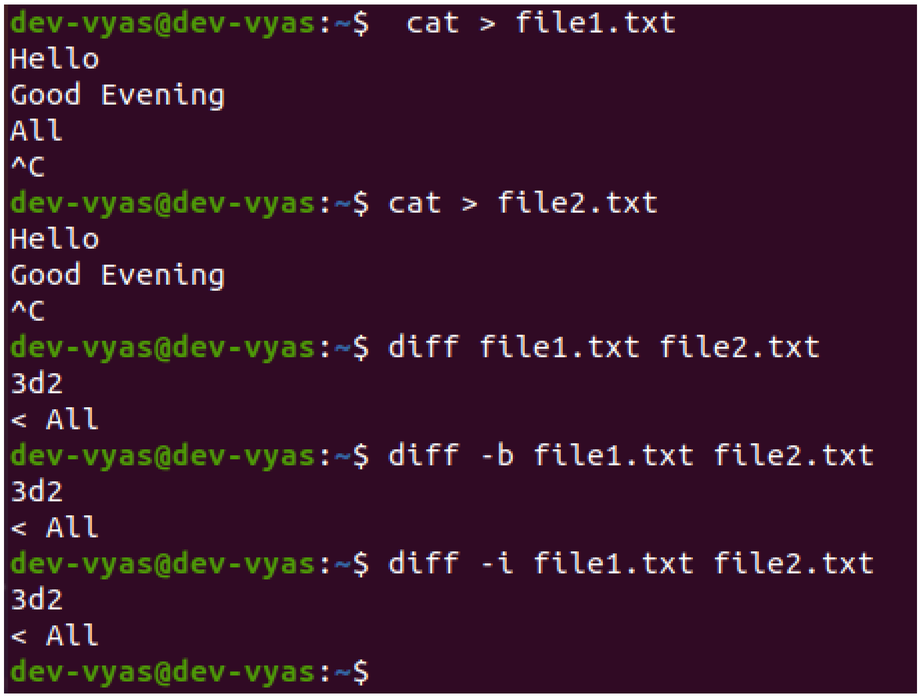
16. diff Command:

- This command is used to display the differences in the files by comparing the files line by line. Diff analyses two files and prints the lines that are different.
- **Syntax:**

  diff [options] File1 File2

- Example:

Option	**Use**
diff -b	Ignores spacing differences
diff –i	Ignores case



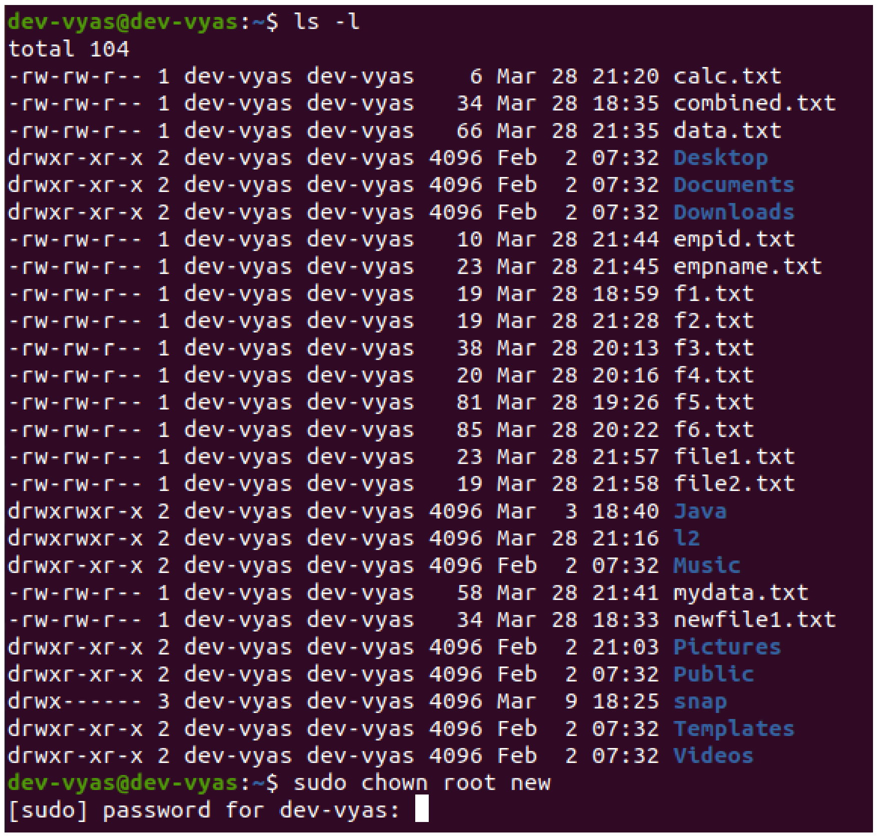
```
dev-vyas@dev-vyas:~$  cat > file1.txt
Hello
Good Evening
All
^C
dev-vyas@dev-vyas:~$ cat > file2.txt
Hello
Good Evening
^C
dev-vyas@dev-vyas:~$ diff file1.txt file2.txt
3d2
< All
dev-vyas@dev-vyas:~$ diff -b file1.txt file2.txt
3d2
< All
dev-vyas@dev-vyas:~$ diff -i file1.txt file2.txt
3d2
< All
dev-vyas@dev-vyas:~$
```

17. chown Command:

- (Change Owner) The chown command changes ownership of files and directories in a Linux filesystem.
- **Syntax:**

  chown [OPTIONS] USER[:GROUP] FILE(s) •

Example:



```
dev-vyas@dev-vyas:~$ ls -l
total 104
-rw-rw-r-- 1 dev-vyas dev-vyas    6 Mar 28 21:20 calc.txt
-rw-rw-r-- 1 dev-vyas dev-vyas   34 Mar 28 18:35 combined.txt
-rw-rw-r-- 1 dev-vyas dev-vyas   66 Mar 28 21:35 data.txt
drwxr-xr-x 2 dev-vyas dev-vyas 4096 Feb  2 07:32 Desktop
drwxr-xr-x 2 dev-vyas dev-vyas 4096 Feb  2 07:32 Documents
drwxr-xr-x 2 dev-vyas dev-vyas 4096 Feb  2 07:32 Downloads
-rw-rw-r-- 1 dev-vyas dev-vyas   10 Mar 28 21:44 empid.txt
-rw-rw-r-- 1 dev-vyas dev-vyas   23 Mar 28 21:45 empname.txt
-rw-rw-r-- 1 dev-vyas dev-vyas   19 Mar 28 18:59 f1.txt
-rw-rw-r-- 1 dev-vyas dev-vyas   19 Mar 28 21:28 f2.txt
-rw-rw-r-- 1 dev-vyas dev-vyas   38 Mar 28 20:13 f3.txt
-rw-rw-r-- 1 dev-vyas dev-vyas   20 Mar 28 20:16 f4.txt
-rw-rw-r-- 1 dev-vyas dev-vyas   81 Mar 28 19:26 f5.txt
-rw-rw-r-- 1 dev-vyas dev-vyas   85 Mar 28 20:22 f6.txt
-rw-rw-r-- 1 dev-vyas dev-vyas   23 Mar 28 21:57 file1.txt
-rw-rw-r-- 1 dev-vyas dev-vyas   19 Mar 28 21:58 file2.txt
drwxrwxr-x 2 dev-vyas dev-vyas 4096 Mar  3 18:40 Java
drwxrwxr-x 2 dev-vyas dev-vyas 4096 Mar 28 21:16 l2
drwxr-xr-x 2 dev-vyas dev-vyas 4096 Feb  2 07:32 Music
-rw-rw-r-- 1 dev-vyas dev-vyas   58 Mar 28 21:41 mydata.txt
-rw-rw-r-- 1 dev-vyas dev-vyas   34 Mar 28 18:33 newfile1.txt
drwxr-xr-x 2 dev-vyas dev-vyas 4096 Feb  2 21:03 Pictures
drwxr-xr-x 2 dev-vyas dev-vyas 4096 Feb  2 07:32 Public
drwx------ 3 dev-vyas dev-vyas 4096 Mar  9 18:25 snap
drwxr-xr-x 2 dev-vyas dev-vyas 4096 Feb  2 07:32 Templates
drwxr-xr-x 2 dev-vyas dev-vyas 4096 Feb  2 07:32 Videos
dev-vyas@dev-vyas:~$ sudo chown root new
[sudo] password for dev-vyas: 
```

18. file Command:

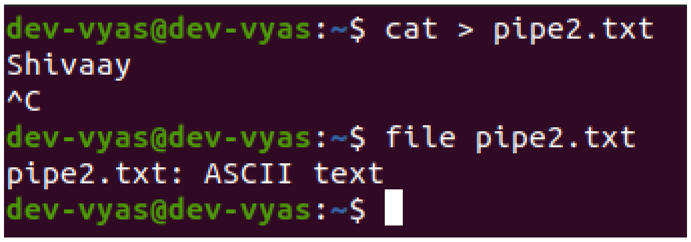
- The file command is used to determine a file's type.
- **Syntax:**

  file [OPTIONS] file1 file2 … •

Example:

Option	Use

file -i	To view the mime type of a file rather than the human readable format

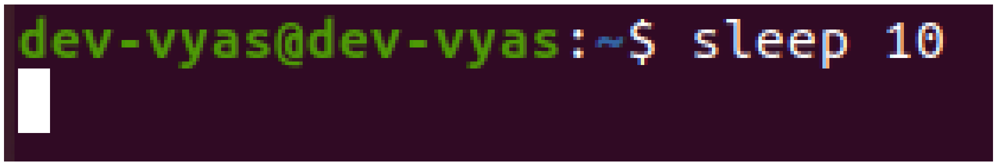


```
dev-vyas@dev-vyas:~$ cat > pipe2.txt
Shivaay
^C
dev-vyas@dev-vyas:~$ file pipe2.txt
pipe2.txt: ASCII text
dev-vyas@dev-vyas:~$
```

19. sleep Command:

- The sleep command is used to delay for a specified amount of time.
- **Syntax:**
  sleep NUMBER[SUFFIX]...
- Example:

Suffix	Use
s	for seconds; this is a default one if you don't specify any letter after the integer.
m	for minutes
h	for hours
d	for days

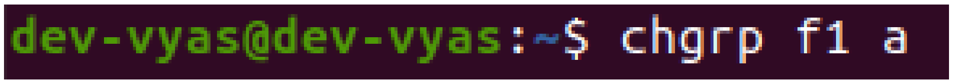


```
dev-vyas@dev-vyas:~$ sleep 10
```

20. chgrp Command:

- (Change Group) The chgrp command is used to change group ownership of a file/directory.
- **Syntax:**
  chgrp [OPTION]… GROUP FILE/DIR… •

Example:

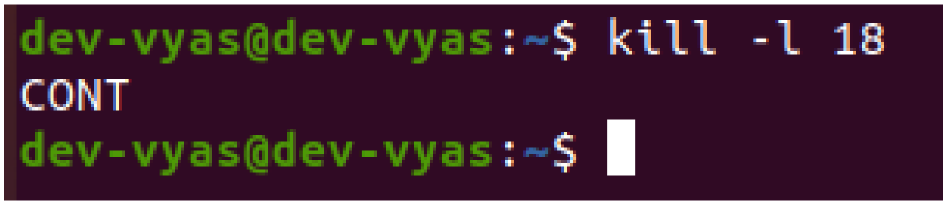


```
dev-vyas@dev-vyas:~$ chgrp f1 a
```

21. kill Command:

- It is used to terminate processes manually.
  Kill command sends a signal to a process which terminates the process. If the user doesn't specify any signal which is to be sent along with kill then default TERM signal is sent that terminates the process..
- **Syntax: kill [option] PID**
- Example:

Option	**Use**
kill -i	To display all the available signals

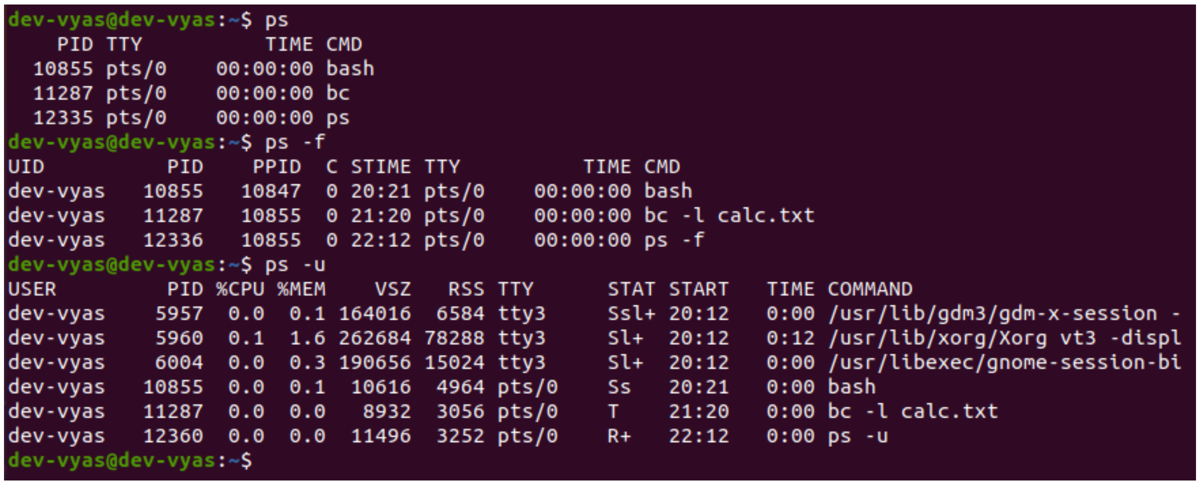


```
dev-vyas@dev-vyas:~$ kill -l 18
CONT
dev-vyas@dev-vyas:~$ 
```

22. ps Command:

- Reports a snapshot of the status of currently running processes.
- **Syntax:**
  ps [option]
- Example:

Option	Use
ps -e	Display every active process on a Linux system in generic (Unix/Linux) format
ps -x	View all processes owned by you
ps -f	To provide more information on processes
ps -u	Filter processes by its user



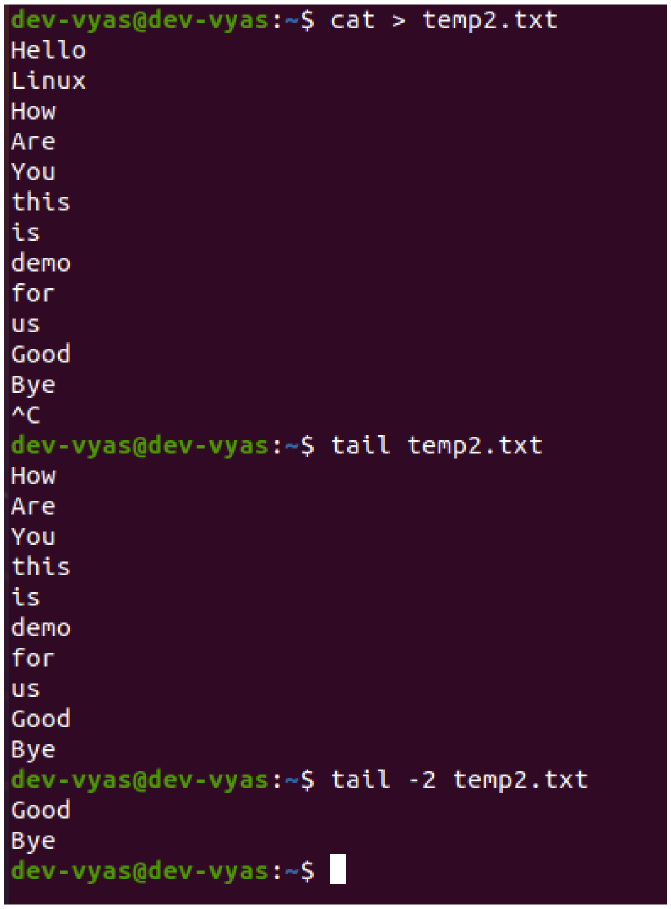
```
dev-vyas@dev-vyas:~$ ps
    PID TTY          TIME CMD
  10855 pts/0    00:00:00 bash
  11287 pts/0    00:00:00 bc
  12335 pts/0    00:00:00 ps
dev-vyas@dev-vyas:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
dev-vyas   10855   10847  0 20:21 pts/0    00:00:00 bash
dev-vyas   11287   10855  0 21:20 pts/0    00:00:00 bc -l calc.txt
dev-vyas   12336   10855  0 22:12 pts/0    00:00:00 ps -f
dev-vyas@dev-vyas:~$ ps -u
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
dev-vyas    5957  0.0  0.1 164016  6584 tty3     Ssl+ 20:12   0:00 /usr/lib/gdm3/gdm-x-session -
dev-vyas    5960  0.1  1.6 262684 78288 tty3     Sl+  20:12   0:12 /usr/lib/xorg/Xorg vt3 -displ
dev-vyas    6004  0.0  0.3 190656 15024 tty3     Sl+  20:12   0:00 /usr/libexec/gnome-session-bi
dev-vyas   10855  0.0  0.1  10616  4964 pts/0    Ss   20:21   0:00 bash
dev-vyas   11287  0.0  0.0   8932  3056 pts/0    T    21:20   0:00 bc -l calc.txt
dev-vyas   12360  0.0  0.0  11496  3252 pts/0    R+   22:12   0:00 ps -u
dev-vyas@dev-vyas:~$
```

23. tail Command:

- tail is a command which prints the last few number of lines (10 lines by default) of a certain file, then terminates.
- **Syntax:** **tail [OPTION]... [FILE]...**

- Example:

Option	Use
tail -n	Output the last num lines, instead of the default (10)
tail -c	Output the last num bytes of each file
tail -q	Never output headers

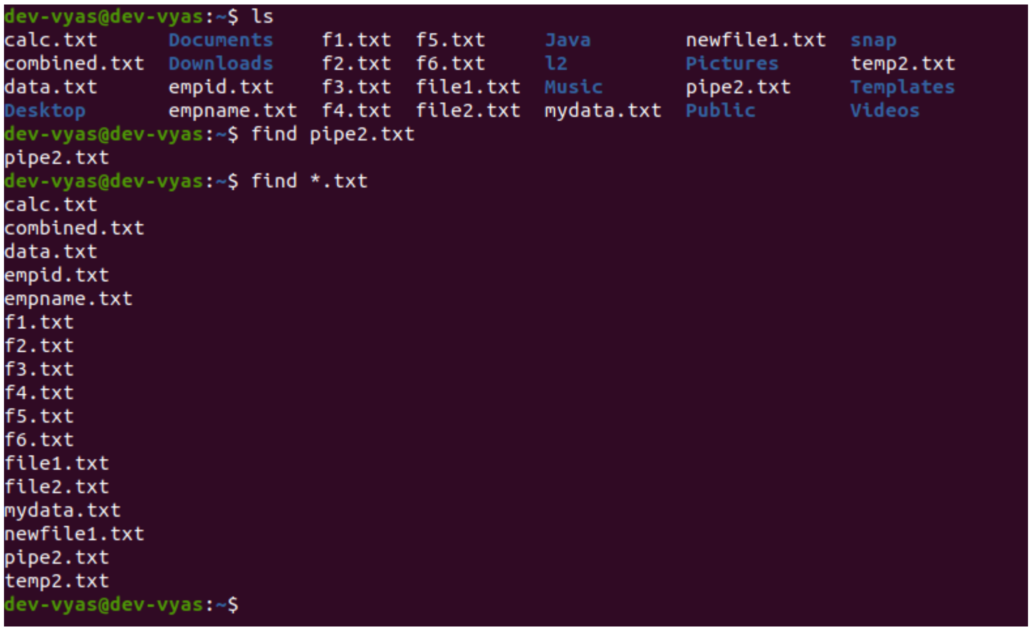


```
dev-vyas@dev-vyas:~$ cat > temp2.txt
Hello
Linux
How
Are
You
this
is
demo
for
us
Good
Bye
^C
dev-vyas@dev-vyas:~$ tail temp2.txt
How
Are
You
this
is
demo
for
us
Good
Bye
dev-vyas@dev-vyas:~$ tail -2 temp2.txt
Good
Bye
dev-vyas@dev-vyas:~$
```

24. find Command:

- find command searches for files in a directory hierarchy.
- **Syntax:**
  find [option] [path...] [expression]
- Example:

Option	Use
find -name filename	Search for files that are specified by ‘filename’
find -newer filename	Search for files that were modified/created after ‘filename’
find -user name	Search for files owned by user name or ID ‘name’
find -size +N/-N	Search for files of ‘N’ blocks; ‘N’ followed by ‘c’ can be used to measure size in characters
find -empty	Search for empty files and directories
find -perm octal	Search for the file if permission is ‘octal’



```
dev-vyas@dev-vyas:~$ ls
calc.txt      Documents    f1.txt  f5.txt     Java        newfile1.txt  snap
combined.txt  Downloads    f2.txt  f6.txt     l2          Pictures      temp2.txt
data.txt      empid.txt    f3.txt  file1.txt  Music       pipe2.txt     Templates
Desktop       empname.txt  f4.txt  file2.txt  mydata.txt  Public        Videos
dev-vyas@dev-vyas:~$ find pipe2.txt
pipe2.txt
dev-vyas@dev-vyas:~$ find *.txt
calc.txt
combined.txt
data.txt
empid.txt
empname.txt
f1.txt
f2.txt
f3.txt
f4.txt
f5.txt
f6.txt
file1.txt
file2.txt
mydata.txt
newfile1.txt
pipe2.txt
temp2.txt
dev-vyas@dev-vyas:~$
```

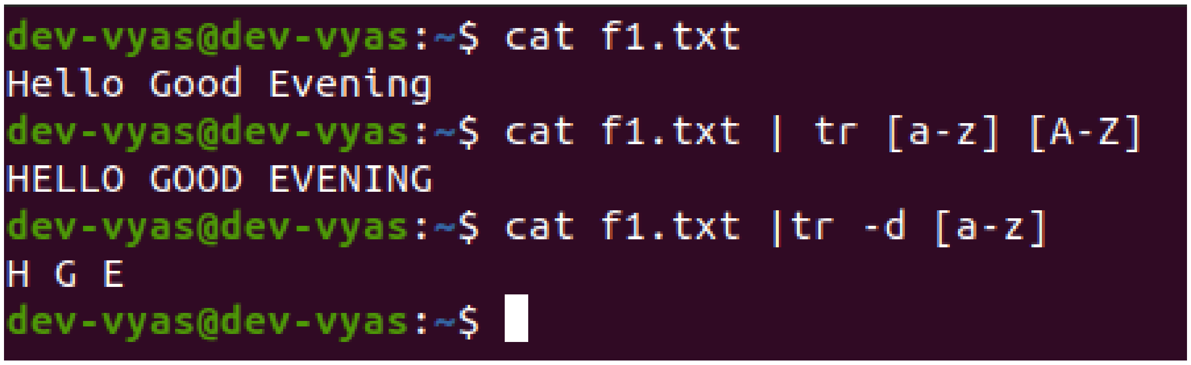
25. tr Command:

- (Translate Command) The tr command in UNIX is a command line utility for translating or deleting characters.
  It supports a range of transformations including uppercase to lowercase, squeezing repeating characters, deleting specific characters and basic find and replace.
  It can be used with UNIX pipes to support more complex translation.
  tr stands for translate.
- **Syntax:**

tr [OPTION] SET1 [SET2] •

Example:

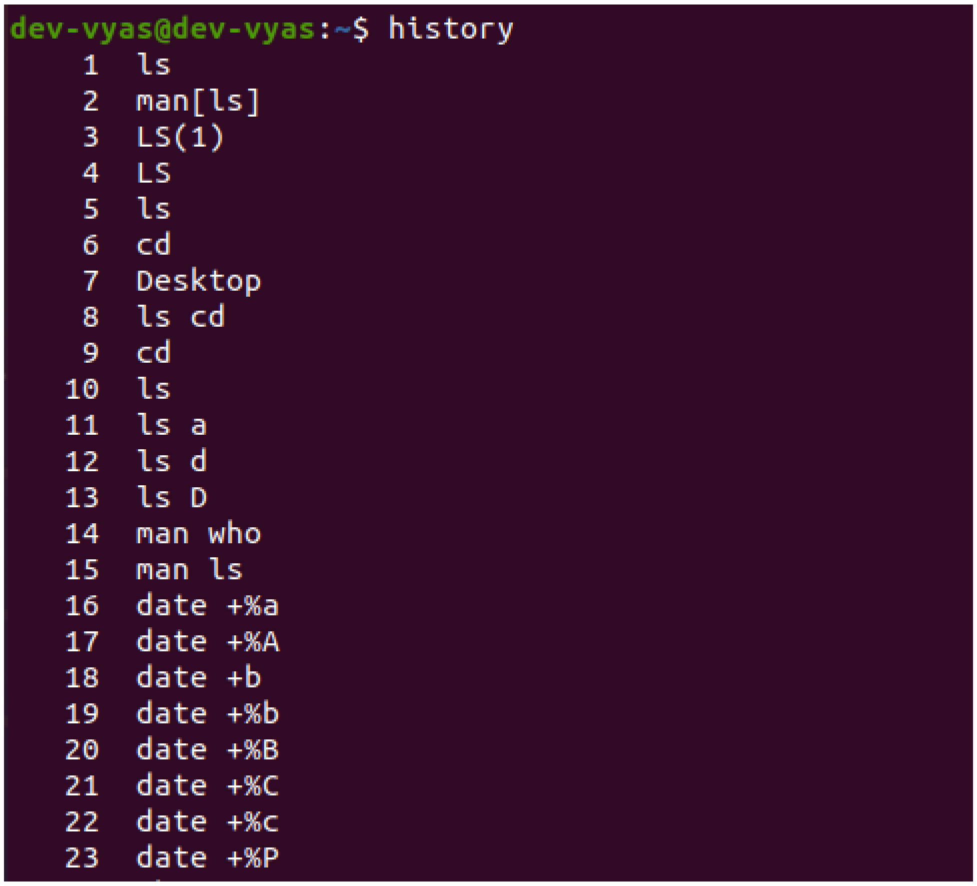
Option	Use
tr -s	Replaces repeated characters listed in the set1 with single occurrence
tr -d	Delete characters in string1 from the input
tr -c	complements the set of characters in string. i.e., operations apply to characters not in the given set
tr -cd	Remove all characters except digits



```
dev-vyas@dev-vyas:~$ cat f1.txt
Hello Good Evening
dev-vyas@dev-vyas:~$ cat f1.txt | tr [a-z] [A-Z]
HELLO GOOD EVENING
dev-vyas@dev-vyas:~$ cat f1.txt |tr -d [a-z]
H G E
dev-vyas@dev-vyas:~$
```

26. history Command:

- history command is used to view the previously executed command.
- **Syntax: History** • Example:



```
dev-vyas@dev-vyas:~$ history
    1  ls
    2  man[ls]
    3  LS(1)
    4  LS
    5  ls
    6  cd
    7  Desktop
    8  ls cd
    9  cd
   10  ls
   11  ls a
   12  ls d
   13  ls D
   14  man who
   15  man ls
   16  date +%a
   17  date +%A
   18  date +b
   19  date +%b
   20  date +%B
   21  date +%C
   22  date +%c
   23  date +%P
```

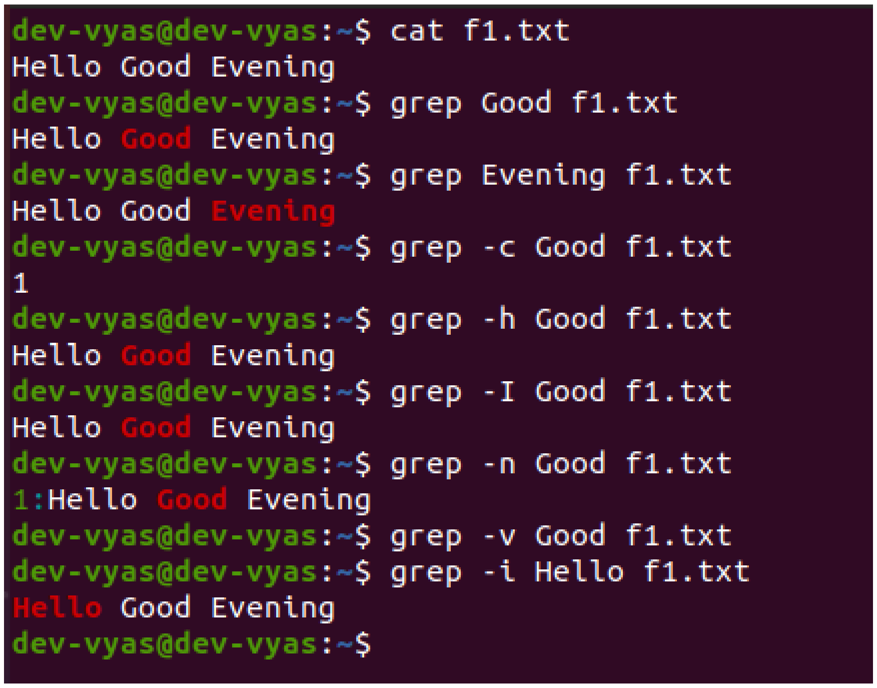
27. grep Command:

- The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern.
  The pattern that is searched in the file is referred to as the regular expression.
  grep stands for globally search for regular expression and print out.
- **Syntax:**
  grep [options] pattern [files] •

Example:

Option	Use
grep -c	Prints only a count of the lines that match a pattern
grep -h	Display the matched lines, but do not display the filenames
grep -I	Displays list of a filenames only
grep -i	Ignores, case for matching
grep -n	Display the matched lines and their line numbers

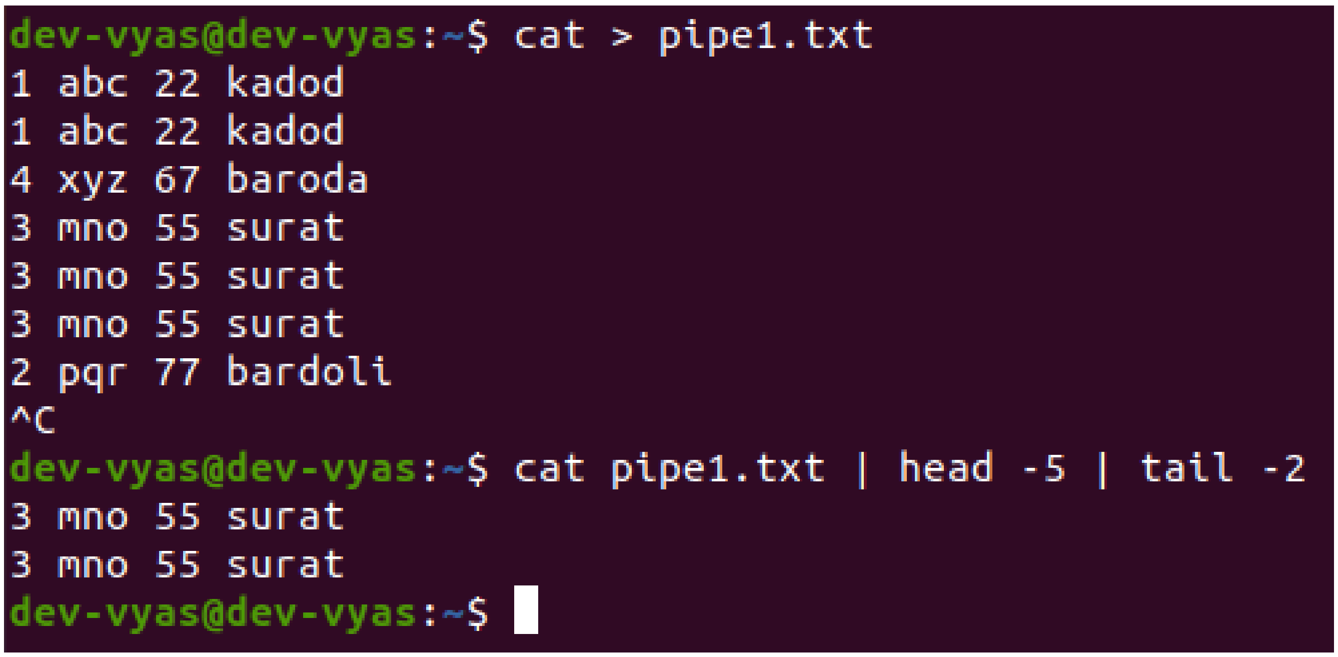
grep -v	This prints out all the lines that do not matches the pattern
grep -w	Match whole word
grep -o	Print only the matched parts of a matching line



```
dev-vyas@dev-vyas:~$ cat f1.txt
Hello Good Evening
dev-vyas@dev-vyas:~$ grep Good f1.txt
Hello Good Evening
dev-vyas@dev-vyas:~$ grep Evening f1.txt
Hello Good Evening
dev-vyas@dev-vyas:~$ grep -c Good f1.txt
1
dev-vyas@dev-vyas:~$ grep -h Good f1.txt
Hello Good Evening
dev-vyas@dev-vyas:~$ grep -I Good f1.txt
Hello Good Evening
dev-vyas@dev-vyas:~$ grep -n Good f1.txt
1:Hello Good Evening
dev-vyas@dev-vyas:~$ grep -v Good f1.txt
dev-vyas@dev-vyas:~$ grep -i Hello f1.txt
Hello Good Evening
dev-vyas@dev-vyas:~$
```

28. pipeline (|) Command:

- It redirects the command STDOUT or standard output into the given next command STDIN or standard input.
  In short, the output of each process directly as input to the next one like a pipeline.
  The symbol '|' denotes a pipe.
  Pipes help you mash-up two or more commands at the same time and run them consecutively.
- **Syntax:**
  command_1 | command_2 | command_3 |..... | command_N…
- Example:

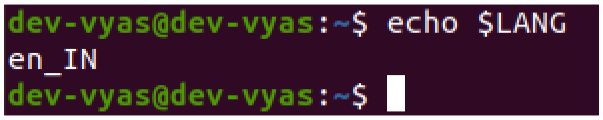


```
dev-vyas@dev-vyas:~$ cat > pipe1.txt
1 abc 22 kadod
1 abc 22 kadod
4 xyz 67 baroda
3 mno 55 surat
3 mno 55 surat
3 mno 55 surat
2 pqr 77 bardoli
^C
dev-vyas@dev-vyas:~$ cat pipe1.txt | head -5 | tail -2
3 mno 55 surat
3 mno 55 surat
dev-vyas@dev-vyas:~$
```

PRACTICAL:-3

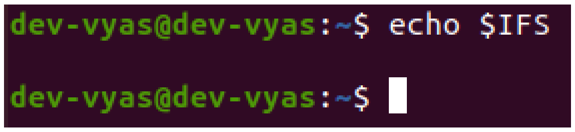
AIM:- Study of UNIX Shell and Environment variables.

1. echo $LANG
   It displays the language used by the system for the user.



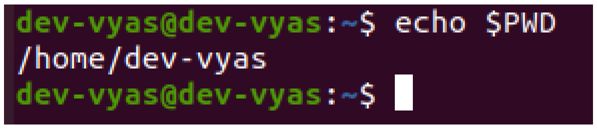
```
dev-vyas@dev-vyas:~$ echo $LANG
en_IN
dev-vyas@dev-vyas:~$
```

2. echo $IFS
   It displays the intermediate field separator used between words. In this system it is wide space.



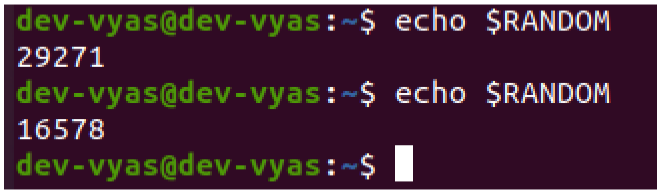
```
dev-vyas@dev-vyas:~$ echo $IFS

dev-vyas@dev-vyas:~$
```

3. echo $PWD
   It displays the present working directory in which user is working.



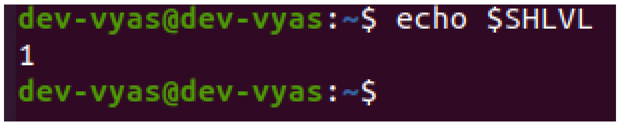
```
dev-vyas@dev-vyas:~$ echo $PWD
/home/dev-vyas
dev-vyas@dev-vyas:~$
```

4. echo $RANDOM
   This command generates or echoes a random number.



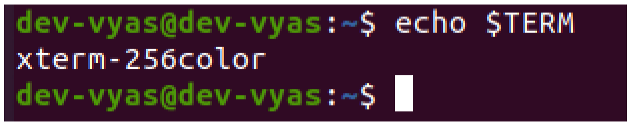
```
dev-vyas@dev-vyas:~$ echo $RANDOM
29271
dev-vyas@dev-vyas:~$ echo $RANDOM
16578
dev-vyas@dev-vyas:~$
```

5. echo $SHLVL
   It shows the shell level in which user is present.



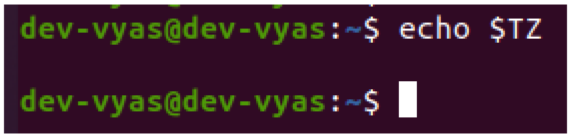
```
dev-vyas@dev-vyas:~$ echo $SHLVL
1
dev-vyas@dev-vyas:~$
```

6. echo $TERM
   It gives terminal information and emulation used by it.



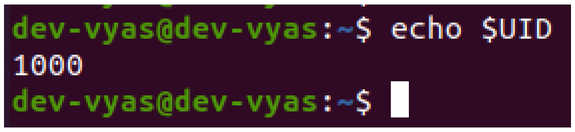
```
dev-vyas@dev-vyas:~$ echo $TERM
xterm-256color
dev-vyas@dev-vyas:~$
```

7. echo $TZ
   It displays time zone set for the system.



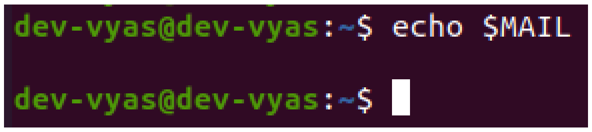
```
dev-vyas@dev-vyas:~$ echo $TZ

dev-vyas@dev-vyas:~$
```

8. echo $UID
   This command shows the user logged in through the id.



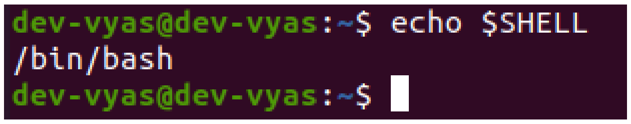
```
dev-vyas@dev-vyas:~$ echo $UID
1000
dev-vyas@dev-vyas:~$
```

9. echo $MAIL
   It shows mailing facility and shows the mails in the mailbox.



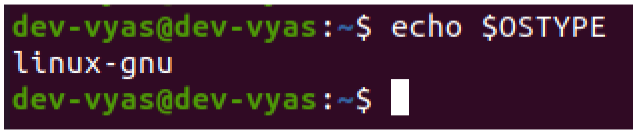
```
dev-vyas@dev-vyas:~$ echo $MAIL

dev-vyas@dev-vyas:~$
```

10. echo $SHELL
    It displays the location of bash of the terminal.



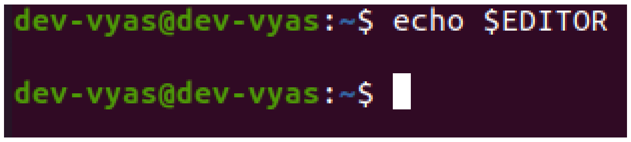
```
dev-vyas@dev-vyas:~$ echo $SHELL
/bin/bash
dev-vyas@dev-vyas:~$
```

11. echo $OSTYPE
    It displays the type of the Operating System.



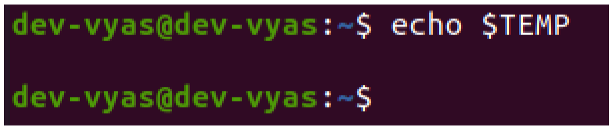
```
dev-vyas@dev-vyas:~$ echo $OSTYPE
linux-gnu
dev-vyas@dev-vyas:~$
```

12. echo $EDITOR
    It displays any default editor of the system.



```
dev-vyas@dev-vyas:~$ echo $EDITOR

dev-vyas@dev-vyas:~$
```

13. echo $TEMP
    It shows all the temporary folders created in the system.



```
dev-vyas@dev-vyas:~$ echo $TEMP

dev-vyas@dev-vyas:~$
```

PRACTICAL:-4

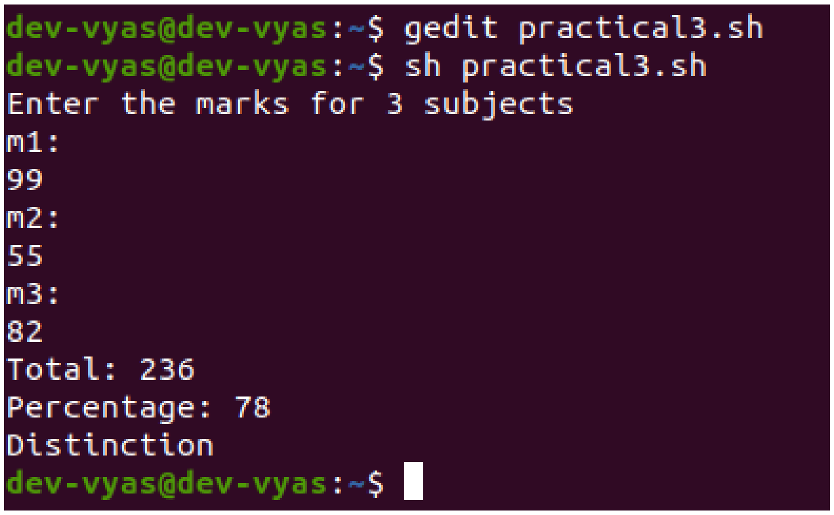
AIM:-Write a shell script to generate marksheet of a student.Take three subjects, calculate and display total marks, percentage and class obtained by the student.

SCRIPT:-

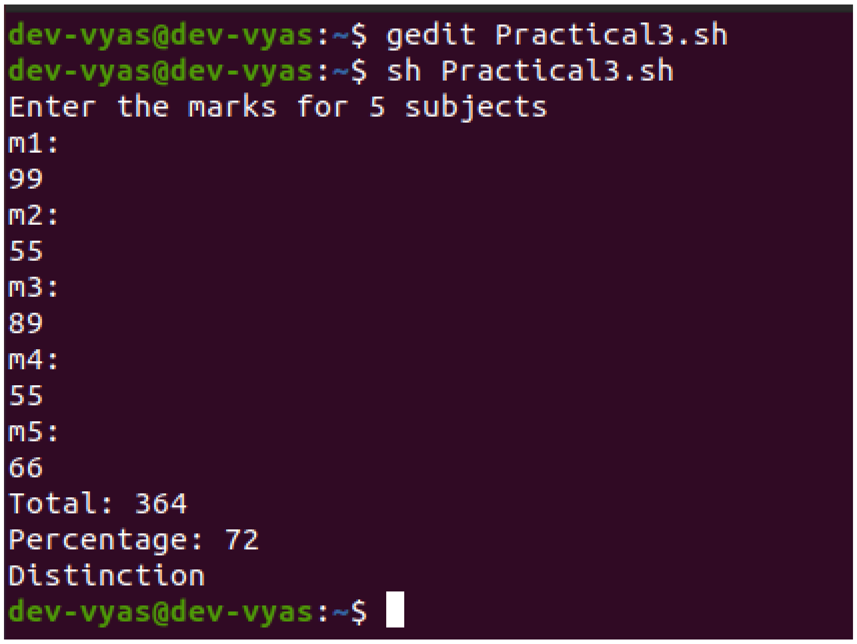
```
echo "Enter the marks for 3 subjects"
echo "m1:"
read m1
echo "m2:"
read m2
echo "m3:"
read m3
sum=`expr $m1 + $m2 + $m3 `
echo "Total:" $sum
per=`expr $sum / 3`
echo "Percentage:" $per
if [ $per -ge 60 ]
then
        echo "Distinction"
elif [ $per -ge 50 ]
then
        echo "First Class"
elif [ $per -ge 40 ]
then
        echo "Second class"
else
        echo "You are failed"
fi
```

```
echo "Enter the marks for 5 subjects"
echo "m1:"
read m1
echo "m2:"
read m2
echo "m3:"
read m3
echo "m4:"
read m4
echo "m5:"
read m5
sum=`expr $m1 + $m2 + $m3 + $m4 + $m5`
echo "Total:" $sum
per=`expr $sum / 5`
echo "Percentage:" $per
if [ $per -ge 60 ]
then
        echo "Distinction"
elif [ $per -ge 50 ]
then
        echo "First Class"
elif [ $per -ge 40 ]
then
        echo "Second class"
else
        echo "You are failed"
fi
```

OUTPUT:-



```
dev-vyas@dev-vyas:~$ gedit practical3.sh
dev-vyas@dev-vyas:~$ sh practical3.sh
Enter the marks for 3 subjects
m1:
99
m2:
55
m3:
82
Total: 236
Percentage: 78
Distinction
dev-vyas@dev-vyas:~$
```



```
dev-vyas@dev-vyas:~$ gedit Practical3.sh
dev-vyas@dev-vyas:~$ sh Practical3.sh
Enter the marks for 5 subjects
m1:
99
m2:
55
m3:
89
m4:
55
m5:
66
Total: 364
Percentage: 72
Distinction
dev-vyas@dev-vyas:~$
```

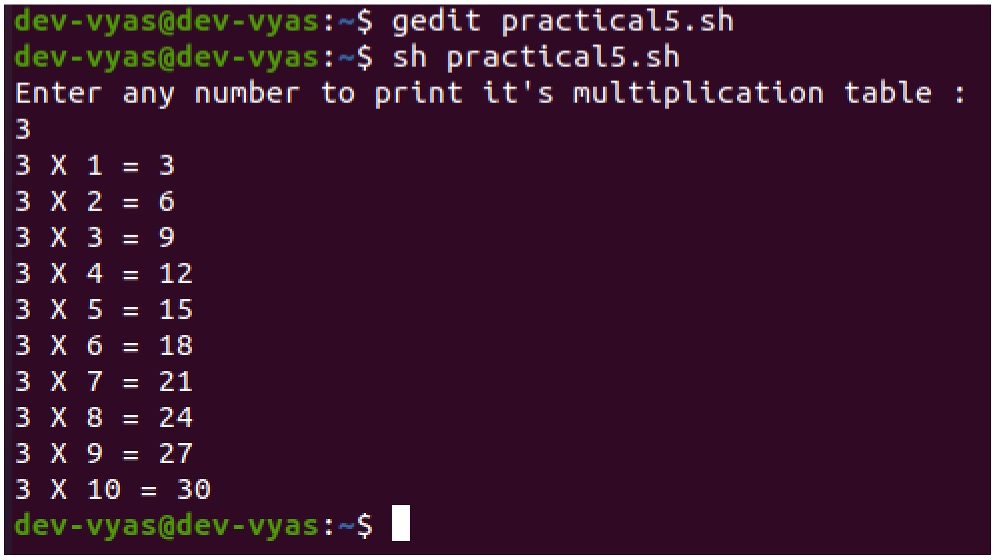
PRACTICAL:-5

AIM:- Write a shell script to display multiplication table of a given number.

SCRIPT:-

```
echo "Enter any number to print it's multiplication table : "

read num

for i in 1 2 3 4 5 6 7 8 9 10

do

        echo "$num X $i = `expr $i \* $num`"

done
```

OUTPUT:-



```
dev-vyas@dev-vyas:~$ gedit practical5.sh
dev-vyas@dev-vyas:~$ sh practical5.sh
Enter any number to print it's multiplication table :
3
3 X 1 = 3
3 X 2 = 6
3 X 3 = 9
3 X 4 = 12
3 X 5 = 15
3 X 6 = 18
3 X 7 = 21
3 X 8 = 24
3 X 9 = 27
3 X 10 = 30
dev-vyas@dev-vyas:~$
```

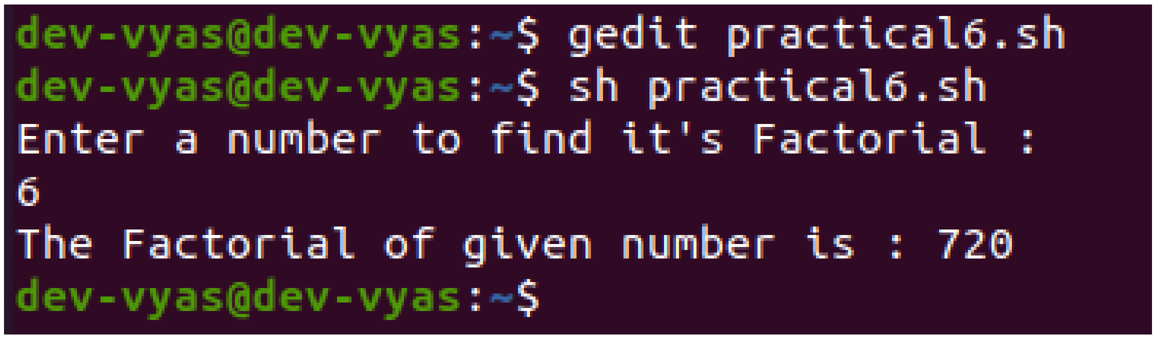
<u>PRACTICAL:-6</u>

<u>AIM:-</u> Write a shell script to find factorial of a given number n.

<u>SCRIPT:-</u>

```
echo "Enter a number to find it's Factorial :"

read num

fact=1

while  [ $num -ge 1 ]

do

        fact=$((fact*num))

        num=$((num-1))

done

echo "The Factorial of given number is : $fact"
```

<u>OUTPUT:-</u>



```
dev-vyas@dev-vyas:~$ gedit practical6.sh
dev-vyas@dev-vyas:~$ sh practical6.sh
Enter a number to find it's Factorial :
6
The Factorial of given number is : 720
dev-vyas@dev-vyas:~$
```

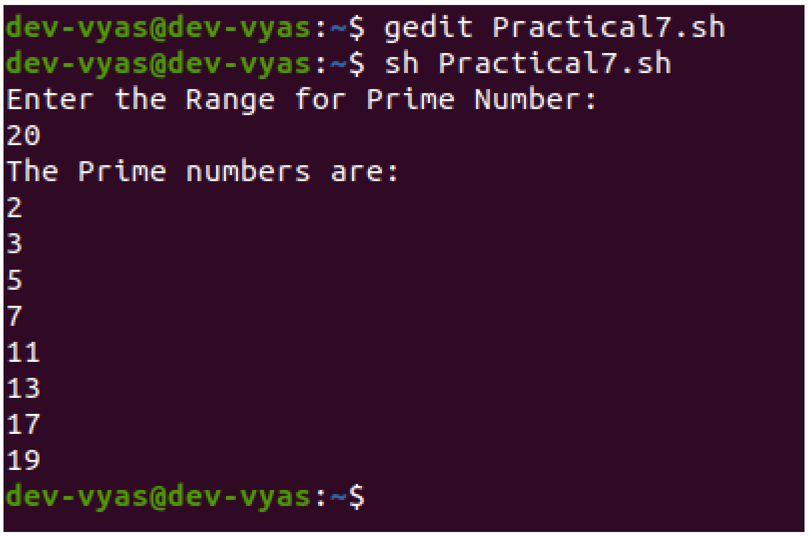
PRACTICAL-7

AIM : Write a shell script which will accept a number b and display first n prime numbers as output.

PROGRAM :

```
echo "Enter the Range for Prime Number:"
read n
echo "The Prime numbers are:"
m=2
while [ $m -le $n ]
do
        i=2
        flag=0
        while [ $i -le `expr $m / 2` ]
        do
                if [ `expr $m % $i` -eq 0 ]
                then
                        flag=1
                        break
                fi
                i=`expr $i + 1`
        done
        if [ $flag -eq 0 ]
        then
                echo $m
        fi
        m=`expr $m + 1`
done
```

OUTPUT :



```
dev-vyas@dev-vyas:~$ gedit Practical7.sh
dev-vyas@dev-vyas:~$ sh Practical7.sh
Enter the Range for Prime Number:
20
The Prime numbers are:
2
3
5
7
11
13
17
19
dev-vyas@dev-vyas:~$
```

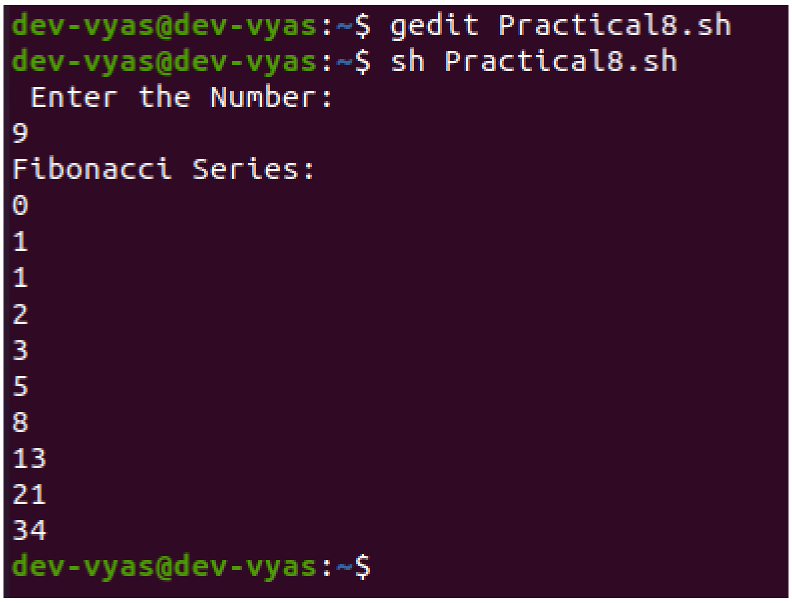
PRACTICAL-8

AIM : Write a shell script which will generate first n Fibonacci numbers like: 1, 1, 2, 3, 5,8,…

PROGRAM :

```
echo " Enter the Number:"
read n
x=0
y=1
i=2
echo "Fibonacci Series:"
echo $x
echo $y
while [ $i -le $n ]
do
        current=`expr $x + $y`
        x=$y
        y=$current
        echo $current
        i=`expr $i + 1`
done
```

OUTPUT :



```
dev-vyas@dev-vyas:~$ gedit Practical8.sh
dev-vyas@dev-vyas:~$ sh Practical8.sh
 Enter the Number:
9
Fibonacci Series:
0
1
1
2
3
5
8
13
21
34
dev-vyas@dev-vyas:~$
```

PRACTICAL-9

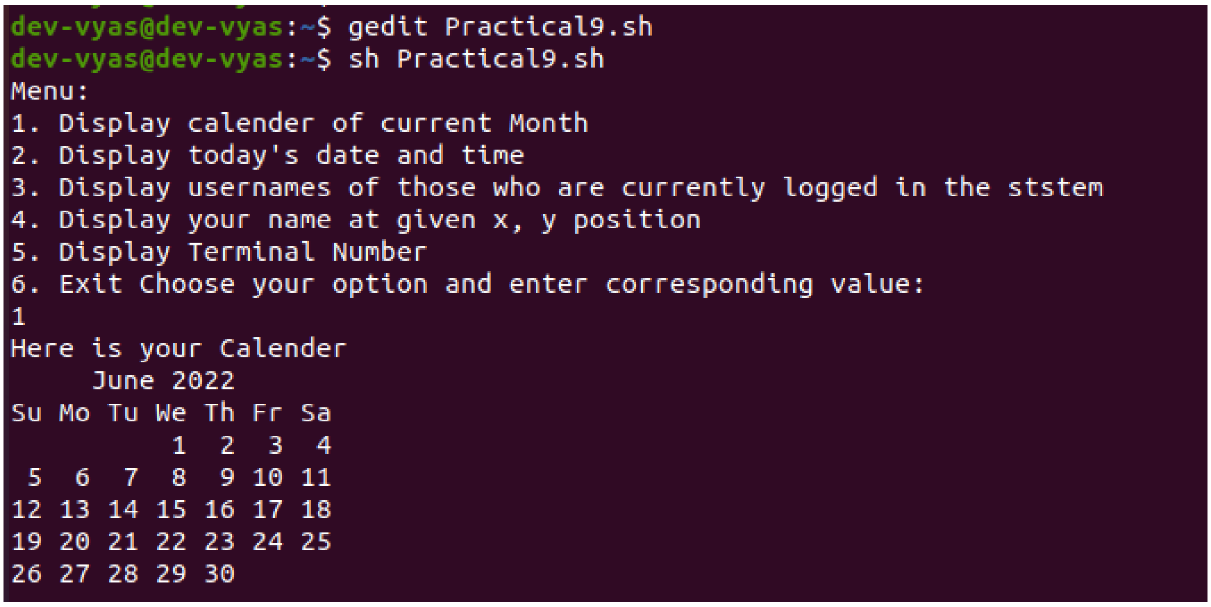
AIM : Write a menu driven shell script which will print the following menu & execute the given task.

1. Display calendar of current month
2. Display today's date and time
3. Display usernames those are currently logged in the system
4. Display your name at given x, y position
5. Display your terminal number

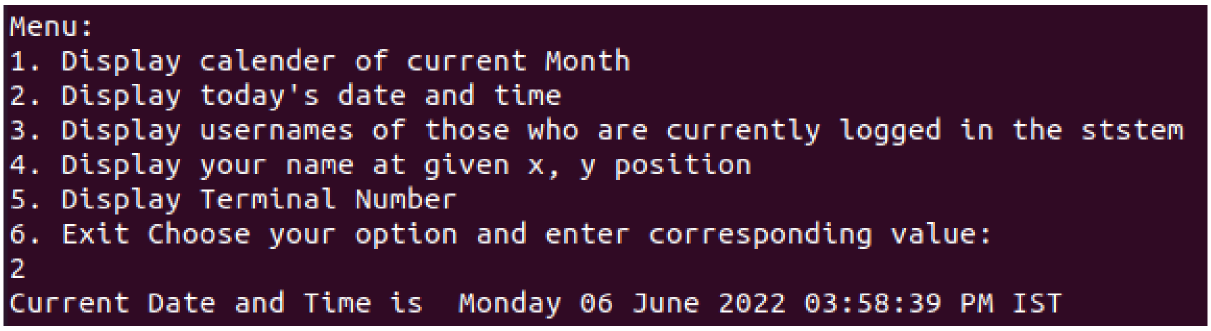
PROGRAM :

```
i=0
while [ $i != 6 ]
do
echo “Menu:
1. Display calender of current Month
2. Display today's date and time
3. Display usernames of those who are currently logged in the ststem
4. Display your name at given x, y position
5. Display Terminal Number
6. Exit Choose your option and Enter Corresponding value:”

read i
case “$i” in
        1) calender="$(cal)"
                echo "Here is your Calender : "
                echo "$calender"
                ;;
         2) current="$(date)"
                echo "Current Date and Time is " "$current"
                ;;
        3) username="$(whoami)"
                echo "Currently logged in users : " “$username”
                ;;
        4)
                ;;
        5)
                ;;
esac
done
```

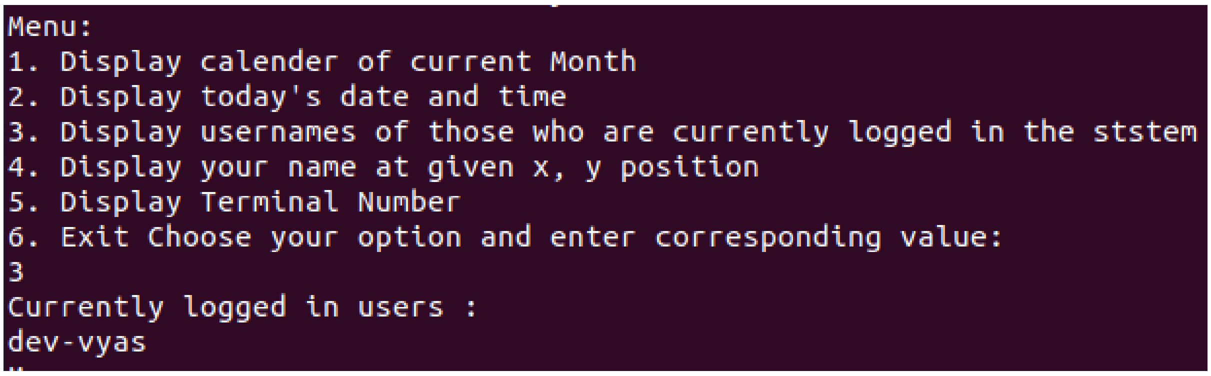
OUTPUT :



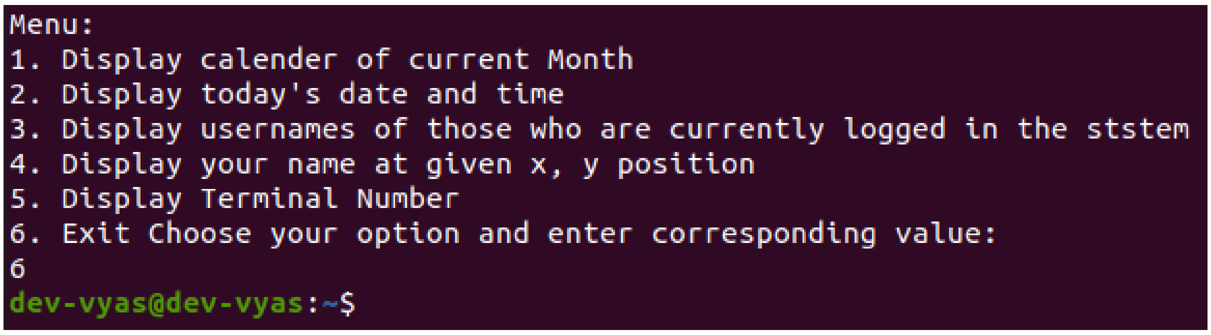
```
dev-vyas@dev-vyas:~$ gedit Practical9.sh
dev-vyas@dev-vyas:~$ sh Practical9.sh
Menu:
1. Display calender of current Month
2. Display today's date and time
3. Display usernames of those who are currently logged in the ststem
4. Display your name at given x, y position
5. Display Terminal Number
6. Exit Choose your option and enter corresponding value:
1
Here is your Calender
     June 2022
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
```



```
Menu:
1. Display calender of current Month
2. Display today's date and time
3. Display usernames of those who are currently logged in the ststem
4. Display your name at given x, y position
5. Display Terminal Number
6. Exit Choose your option and enter corresponding value:
2
Current Date and Time is  Monday 06 June 2022 03:58:39 PM IST
```



```
Menu:
1. Display calender of current Month
2. Display today's date and time
3. Display usernames of those who are currently logged in the ststem
4. Display your name at given x, y position
5. Display Terminal Number
6. Exit Choose your option and enter corresponding value:
3
Currently logged in users :
dev-vyas
```



```
Menu:
1. Display calender of current Month
2. Display today's date and time
3. Display usernames of those who are currently logged in the ststem
4. Display your name at given x, y position
5. Display Terminal Number
6. Exit Choose your option and enter corresponding value:
6
dev-vyas@dev-vyas:~$
```

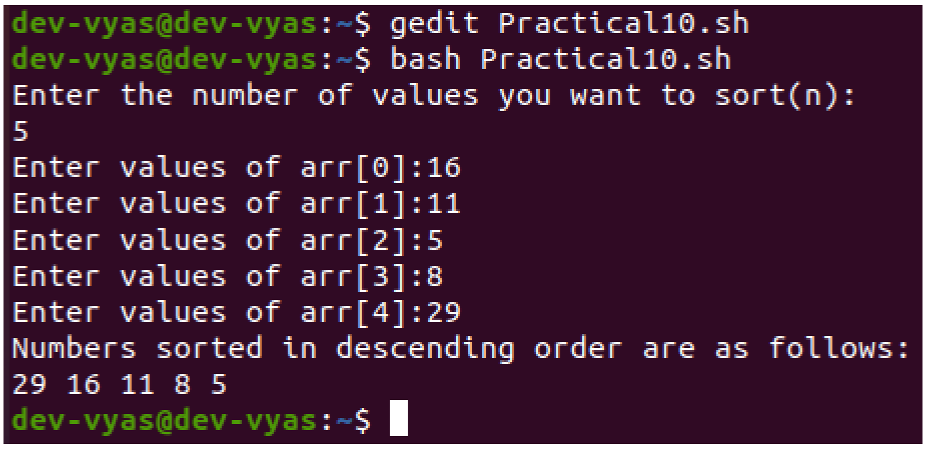
PRACTICAL-10

AIM : Write a shell script to read n numbers as command arguments and sort them in descending order.

PROGRAM:

```
count=0
arr=()
for i in $*
do

        arr[$count]=$i
        count=`expr $count + 1`
done

len=$count

while [ $count -ge 0 ]
do
        temp=0
        while [ $temp -le $count ]
        do
                if [[ ${arr[$temp]} -le ${arr[`expr $temp + 1 `]} ]]
                then
                        temp2=${arr[$temp]}
                        arr[$temp]=${arr[`expr $temp + 1`]}
                        arr[`expr $temp + 1`]=$temp2
                fi
                temp=`expr $temp + 1`
        done
        count=`expr $count - 1`
done
temp=0
while [ $temp -le $len ]
do
        echo ${arr[$temp]}
        temp=`expr $temp + 1`
done
```

OUTPUT:



```
dev-vyas@dev-vyas:~$ gedit Practical10.sh
dev-vyas@dev-vyas:~$ bash Practical10.sh
Enter the number of values you want to sort(n):
5
Enter values of arr[0]:16
Enter values of arr[1]:11
Enter values of arr[2]:5
Enter values of arr[3]:8
Enter values of arr[4]:29
Numbers sorted in descending order are as follows:
29 16 11 8 5
dev-vyas@dev-vyas:~$ 
```

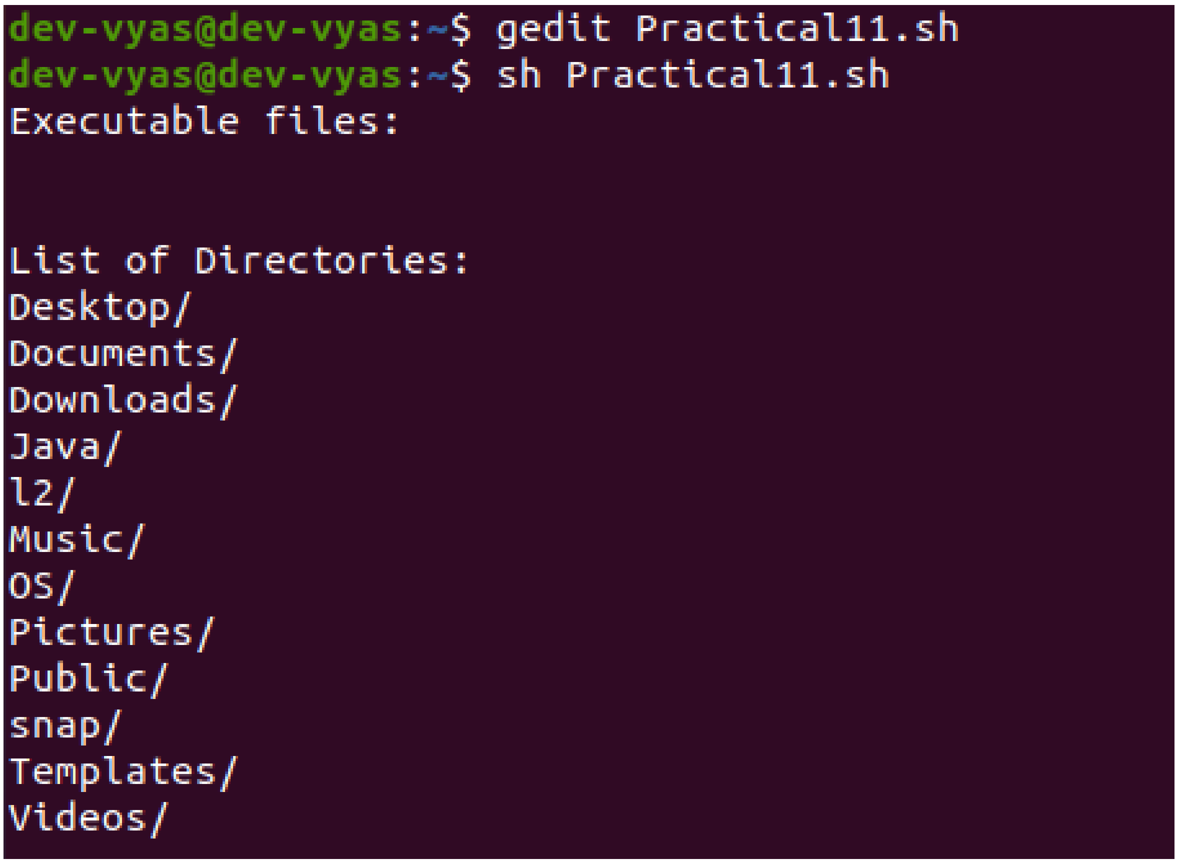
PRACTICAL-11

AIM : Write a shell script to display all executable files, directories and zero sized files from current directory.

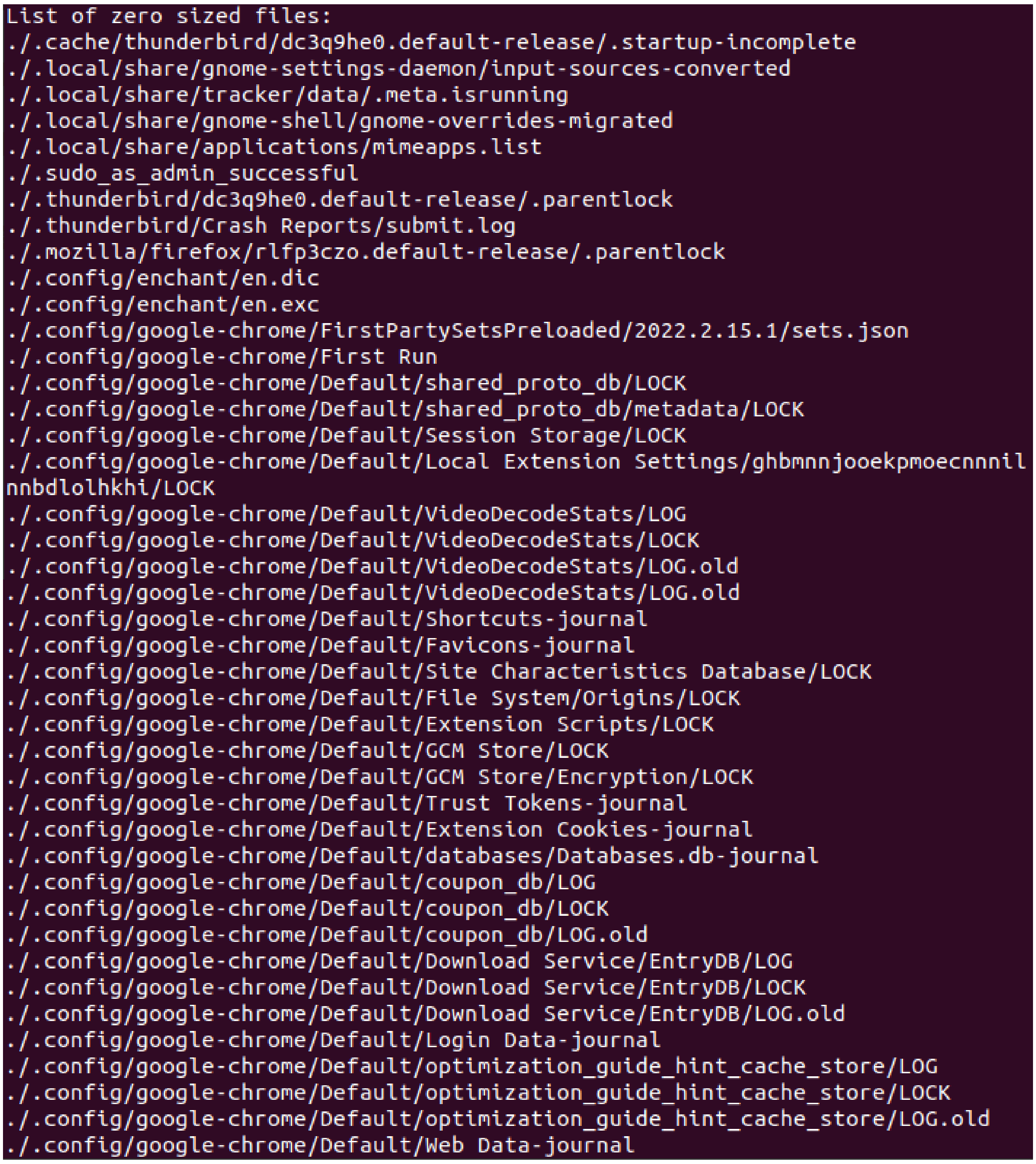
PROGRAM:

```
echo "Executable files: "
files="$(find OS -executable -type f)"
echo "$files"
echo
echo "List of Directories: "
dir="$(ls -d */)"
echo "$dir"
echo
echo "List of zero sized files: "
zero="$(find -size 0)"
echo "$zero"
```

OUTPUT:



```
dev-vyas@dev-vyas:~$ gedit Practical11.sh
dev-vyas@dev-vyas:~$ sh Practical11.sh
Executable files:


List of Directories:
Desktop/
Documents/
Downloads/
Java/
l2/
Music/
OS/
Pictures/
Public/
snap/
Templates/
Videos/
```



```
List of zero sized files:
./.cache/thunderbird/dc3q9he0.default-release/.startup-incomplete
./.local/share/gnome-settings-daemon/input-sources-converted
./.local/share/tracker/data/.meta.isrunning
./.local/share/gnome-shell/gnome-overrides-migrated
./.local/share/applications/mimeapps.list
./.sudo_as_admin_successful
./.thunderbird/dc3q9he0.default-release/.parentlock
./.thunderbird/Crash Reports/submit.log
./.mozilla/firefox/rlfp3czo.default-release/.parentlock
./.config/enchant/en.dic
./.config/enchant/en.exc
./.config/google-chrome/FirstPartySetsPreloaded/2022.2.15.1/sets.json
./.config/google-chrome/First Run
./.config/google-chrome/Default/shared_proto_db/LOCK
./.config/google-chrome/Default/shared_proto_db/metadata/LOCK
./.config/google-chrome/Default/Session Storage/LOCK
./.config/google-chrome/Default/Local Extension Settings/ghbmnnjooekpmoecnnnil
nnbdlolhkhi/LOCK
./.config/google-chrome/Default/VideoDecodeStats/LOG
./.config/google-chrome/Default/VideoDecodeStats/LOCK
./.config/google-chrome/Default/VideoDecodeStats/LOG.old
./.config/google-chrome/Default/VideoDecodeStats/LOG.old
./.config/google-chrome/Default/Shortcuts-journal
./.config/google-chrome/Default/Favicons-journal
./.config/google-chrome/Default/Site Characteristics Database/LOCK
./.config/google-chrome/Default/File System/Origins/LOCK
./.config/google-chrome/Default/Extension Scripts/LOCK
./.config/google-chrome/Default/GCM Store/LOCK
./.config/google-chrome/Default/GCM Store/Encryption/LOCK
./.config/google-chrome/Default/Trust Tokens-journal
./.config/google-chrome/Default/Extension Cookies-journal
./.config/google-chrome/Default/databases/Databases.db-journal
./.config/google-chrome/Default/coupon_db/LOG
./.config/google-chrome/Default/coupon_db/LOCK
./.config/google-chrome/Default/coupon_db/LOG.old
./.config/google-chrome/Default/Download Service/EntryDB/LOG
./.config/google-chrome/Default/Download Service/EntryDB/LOCK
./.config/google-chrome/Default/Download Service/EntryDB/LOG.old
./.config/google-chrome/Default/Login Data-journal
./.config/google-chrome/Default/optimization_guide_hint_cache_store/LOG
./.config/google-chrome/Default/optimization_guide_hint_cache_store/LOCK
./.config/google-chrome/Default/optimization_guide_hint_cache_store/LOG.old
./.config/google-chrome/Default/Web Data-journal
```

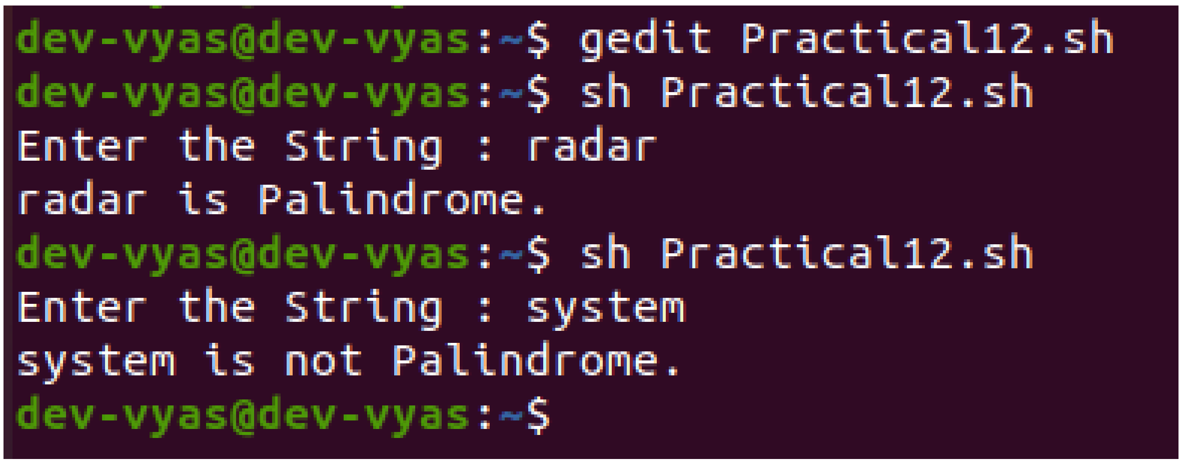
PRACTICAL-12

AIM : Write a shell script to check entered string is palindrome or not.

PROGRAM:

```
read -p "Enter the String : " s
echo $s>temp
rvs="$(rev temp)"
if [ $s = $rvs ]
then
echo "$s is Palindrome."
else
echo "$s is not Palindrome."
fI
```

OUTPUT:



```
dev-vyas@dev-vyas:~$ gedit Practical12.sh
dev-vyas@dev-vyas:~$ sh Practical12.sh
Enter the String : radar
radar is Palindrome.
dev-vyas@dev-vyas:~$ sh Practical12.sh
Enter the String : system
system is not Palindrome.
dev-vyas@dev-vyas:~$
```

PRACTICAL-13

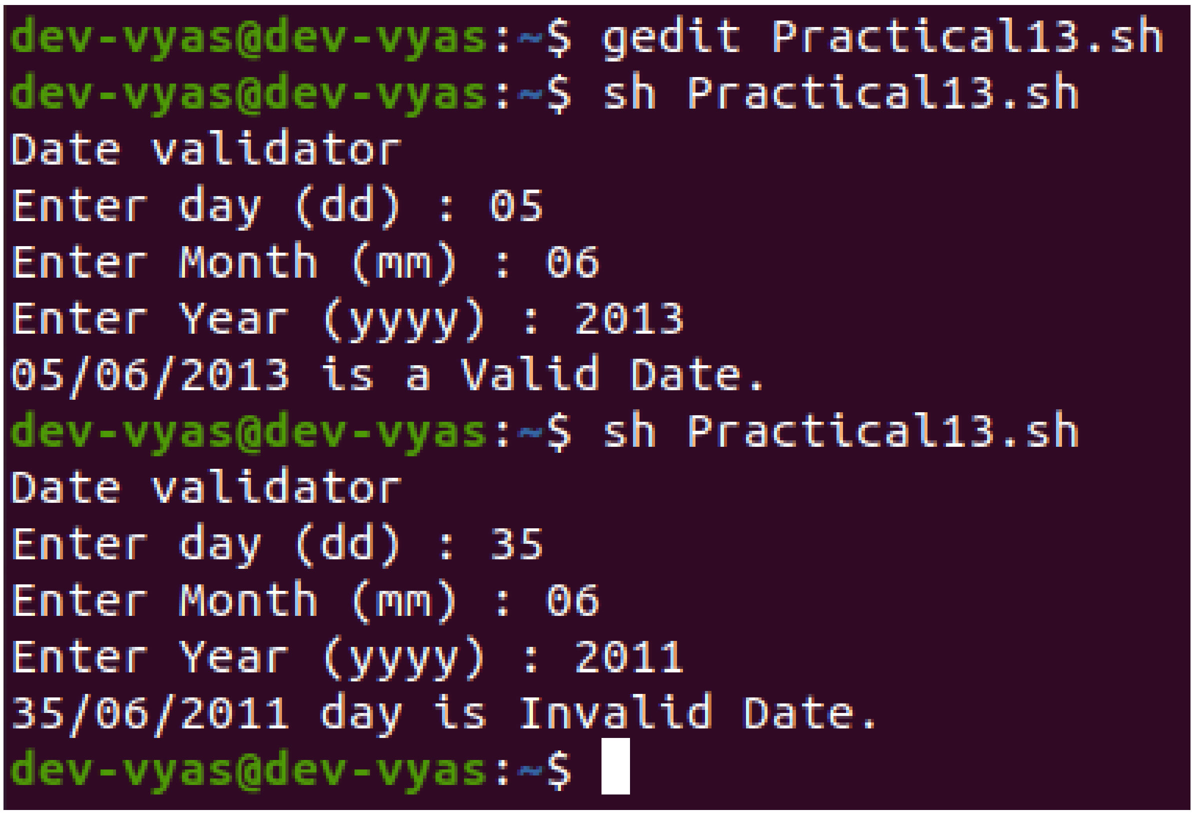
AIM : Write a shell script to validate the entered date. (eg. Date format is : dd-mm-yyyy).

PROGRAM:

```
echo "Date validator"
dd=0
mm=0
yy=0
days=0

read -p "Enter day (dd) : " dd
read -p "Enter Month (mm) : " mm
read -p "Enter Year (yyyy) : " yy

if [ $mm -le 0 -o $mm -gt 12 ]
then
        echo "$mm is Invalid Month. "
        exit 1
fi

case $mm in
        01|03|05|07|08|10|12)
        days=31
        ;;
02)
        days=28
        ;;
        04|06|09|11)
        days=30
        ;;
*)
        days=-1
        ;;
esac

if [ $mm -eq 2 ]
then
        a=`expr $yy % 4`
        b=`expr $yy % 100`
        c=`expr $yy % 400`

if [ $a -eq 0 -a $b -ne 0 -o $c -eq 0 ]
then
        days=29
        else
        break
fi
```

```
fi

if [ $dd -le 0 -o $dd -gt $days ]
then
        echo "$dd/$mm/$yy day is Invalid Date."
exit 3
fi

echo "$dd/$mm/$yy is a Valid Date."
```

OUTPUT:



```
dev-vyas@dev-vyas:~$ gedit Practical13.sh
dev-vyas@dev-vyas:~$ sh Practical13.sh
Date validator
Enter day (dd) : 05
Enter Month (mm) : 06
Enter Year (yyyy) : 2013
05/06/2013 is a Valid Date.
dev-vyas@dev-vyas:~$ sh Practical13.sh
Date validator
Enter day (dd) : 35
Enter Month (mm) : 06
Enter Year (yyyy) : 2011
35/06/2011 day is Invalid Date.
dev-vyas@dev-vyas:~$ 
```

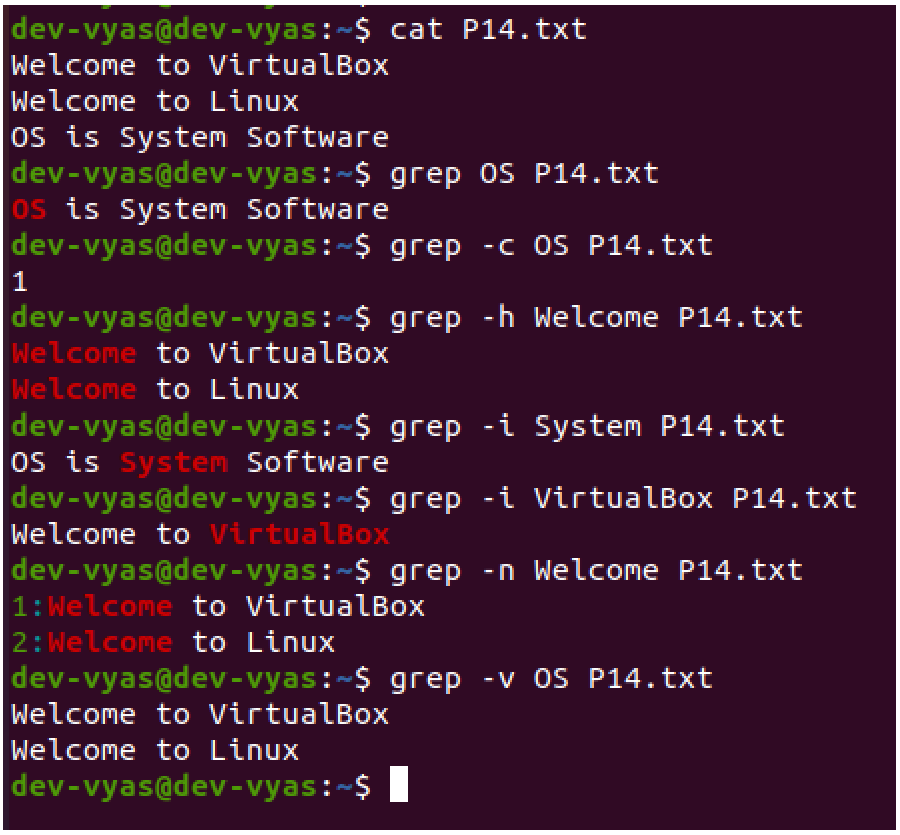
PRACTICAL-14

AIM : Write a shell script to find a word inside of a file using grep, egrep , fgrep.

1.) grep:

- The grep filter searches a file for a particular pattern of characters, and displays all linesthat contain that pattern.
- The pattern that is searched in the file is referred to as the regular expression (grep standsfor global search for regular expression and print out).
- **Syntax** : grep [options] pattern [files]

Option	**Use**
-c	This prints only a count of the lines that match a pattern.
-h	Display the matched lines,but do not display the filenames.
-i	Ignores,case for matching.
-l	Displays list of a filenames only.
-n	Display the matched lines and their line numbers.
-v	This prints out all the lines that do not matches the pattern.

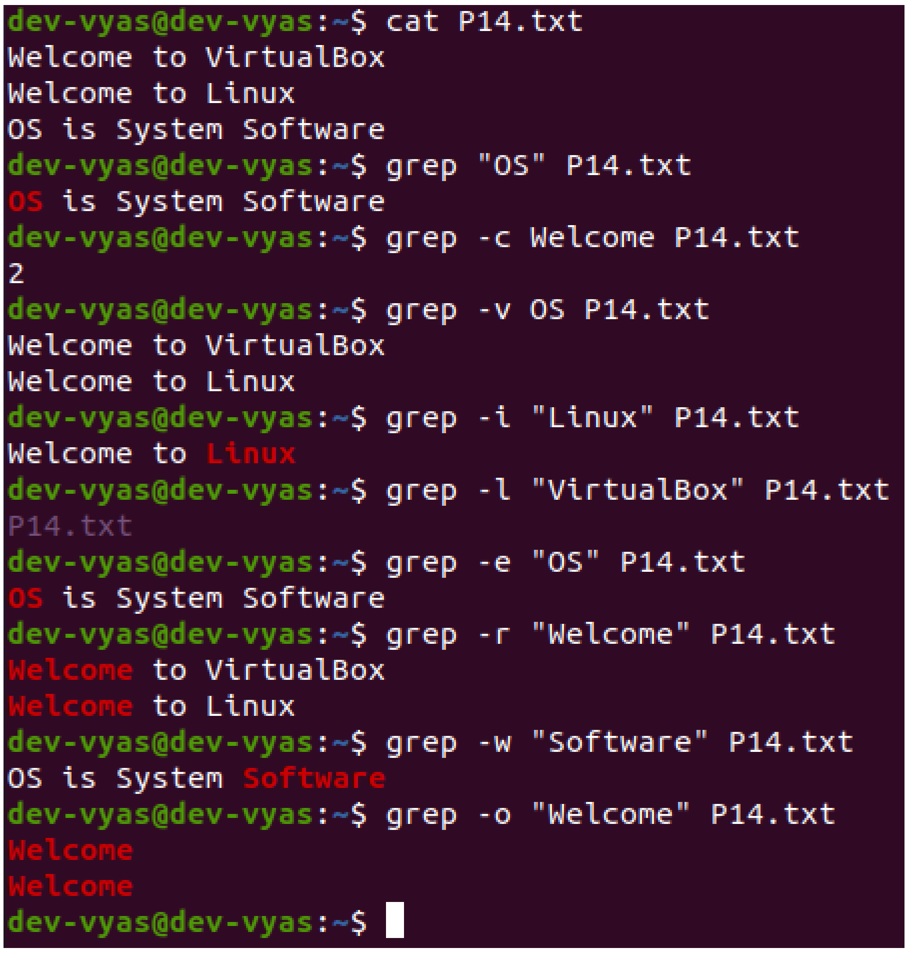


```
dev-vyas@dev-vyas:~$ cat P14.txt
Welcome to VirtualBox
Welcome to Linux
OS is System Software
dev-vyas@dev-vyas:~$ grep OS P14.txt
OS is System Software
dev-vyas@dev-vyas:~$ grep -c OS P14.txt
1
dev-vyas@dev-vyas:~$ grep -h Welcome P14.txt
Welcome to VirtualBox
Welcome to Linux
dev-vyas@dev-vyas:~$ grep -i System P14.txt
OS is System Software
dev-vyas@dev-vyas:~$ grep -i VirtualBox P14.txt
Welcome to VirtualBox
dev-vyas@dev-vyas:~$ grep -n Welcome P14.txt
1:Welcome to VirtualBox
2:Welcome to Linux
dev-vyas@dev-vyas:~$ grep -v OS P14.txt
Welcome to VirtualBox
Welcome to Linux
dev-vyas@dev-vyas:~$
```

2.) egrep:

- egrep is a pattern searching command which belongs to the family of grep functions.
- It treats the pattern as an extended regular expression and prints out the lines that match thepattern.
- **Syntax** : egrep [options] pattern [files]

Option	**Use**
-c	Used to counts and prints the number of lines that matches the pattern and not the lines.
-v	It prints the lines that does not match with the pattern.
-i	Ignore the case of the pattern while matching.
-l	Prints only the names of the files that matched.
-L	Prints only the names of the files that did not have the pattern Ooposite of -l flag.

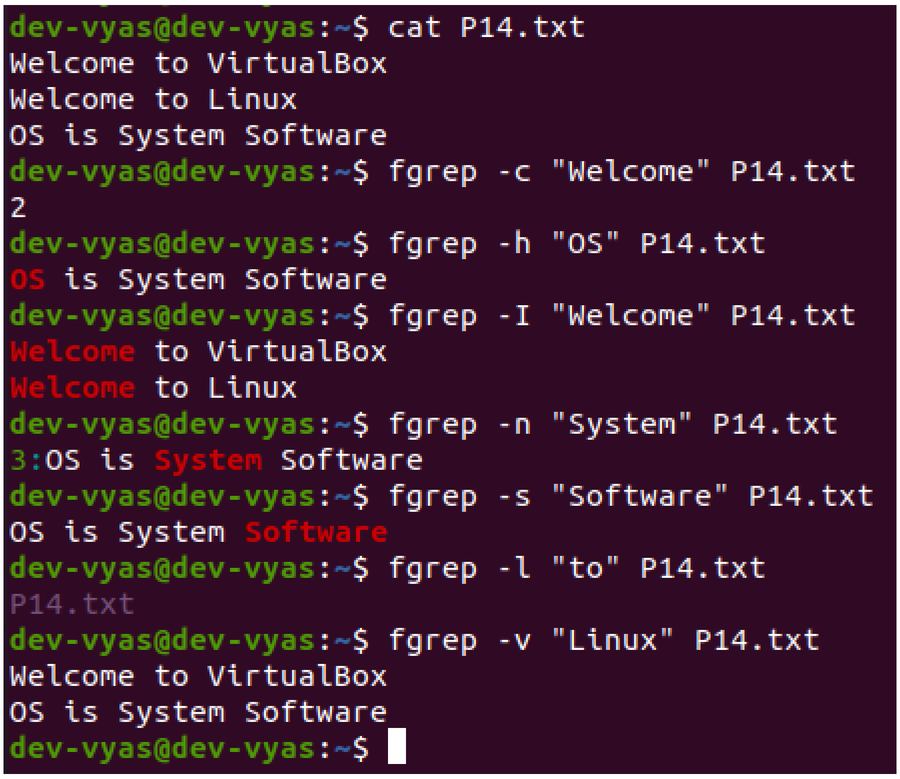


```
dev-vyas@dev-vyas:~$ cat P14.txt
Welcome to VirtualBox
Welcome to Linux
OS is System Software
dev-vyas@dev-vyas:~$ grep "OS" P14.txt
OS is System Software
dev-vyas@dev-vyas:~$ grep -c Welcome P14.txt
2
dev-vyas@dev-vyas:~$ grep -v OS P14.txt
Welcome to VirtualBox
Welcome to Linux
dev-vyas@dev-vyas:~$ grep -i "Linux" P14.txt
Welcome to Linux
dev-vyas@dev-vyas:~$ grep -l "VirtualBox" P14.txt
P14.txt
dev-vyas@dev-vyas:~$ grep -e "OS" P14.txt
OS is System Software
dev-vyas@dev-vyas:~$ grep -r "Welcome" P14.txt
Welcome to VirtualBox
Welcome to Linux
dev-vyas@dev-vyas:~$ grep -w "Software" P14.txt
OS is System Software
dev-vyas@dev-vyas:~$ grep -o "Welcome" P14.txt
Welcome
Welcome
dev-vyas@dev-vyas:~$ 
```

3.) fgrep:

- The fgrep filter is used to search for the fixed-character strings in a file.
- There can be multiple files also to be searched.
- This command is useful when you need to search for strings which contain lots of regularexpression metacharacters, such as “^”, “$”, etc.
- **Syntax** : fgrep [-e pattern_list] [pattern] [file]

Option	**Use**
-c	It is used to print only a count of the lines which contain the pattern.
-h	Used to display the matched lines.
-i	During comparisons,it will ignore upper/lower case distinction.
-n	It is used precede each line by its line number in the file (file line is 1).
-s	It will only display the error messages.



```
dev-vyas@dev-vyas:~$ cat P14.txt
Welcome to VirtualBox
Welcome to Linux
OS is System Software
dev-vyas@dev-vyas:~$ fgrep -c "Welcome" P14.txt
2
dev-vyas@dev-vyas:~$ fgrep -h "OS" P14.txt
OS is System Software
dev-vyas@dev-vyas:~$ fgrep -I "Welcome" P14.txt
Welcome to VirtualBox
Welcome to Linux
dev-vyas@dev-vyas:~$ fgrep -n "System" P14.txt
3:OS is System Software
dev-vyas@dev-vyas:~$ fgrep -s "Software" P14.txt
OS is System Software
dev-vyas@dev-vyas:~$ fgrep -l "to" P14.txt
P14.txt
dev-vyas@dev-vyas:~$ fgrep -v "Linux" P14.txt
Welcome to VirtualBox
OS is System Software
dev-vyas@dev-vyas:~$
```

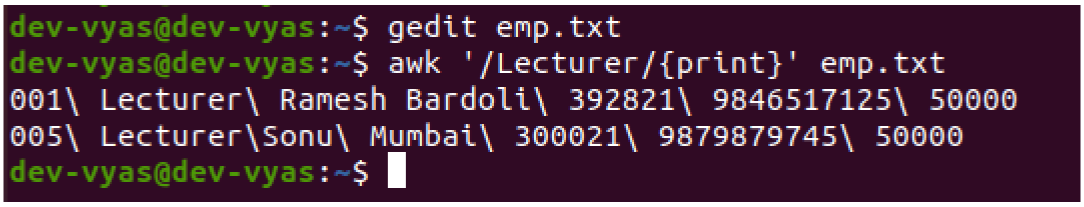
PRACTICAL-15

AIM : Study and Execute AWK script for all following Concepts.

Database (Employee_ID, Post, Employee_Name, Add, PIN, PH_No, Salary)

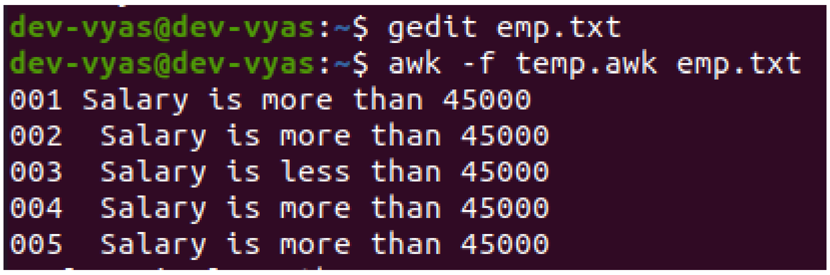
001\ Lecturer\ Ramesh Bardoli\ 392821\ 9846517125\ 50000
002\ HOD\Lokesh\ Surat\ 354698\ 8795157164\ 60000
003\ Lab Assistant\Lily\ Surat\ 354699\ 9871819846\ 40000
004\ Administrator\Gogi\ Baroda\ 316489\ 89797856451\ 45000
005\ Lecturer\Sonu\ Mumbai\ 300021\ 9879879745\ 50000

1. Use of System Variables.



```
dev-vyas@dev-vyas:~$ gedit emp.txt
dev-vyas@dev-vyas:~$ awk '/Lecturer/{print}' emp.txt
001\ Lecturer\ Ramesh Bardoli\ 392821\ 9846517125\ 50000
005\ Lecturer\Sonu\ Mumbai\ 300021\ 9879879745\ 50000
dev-vyas@dev-vyas:~$
```

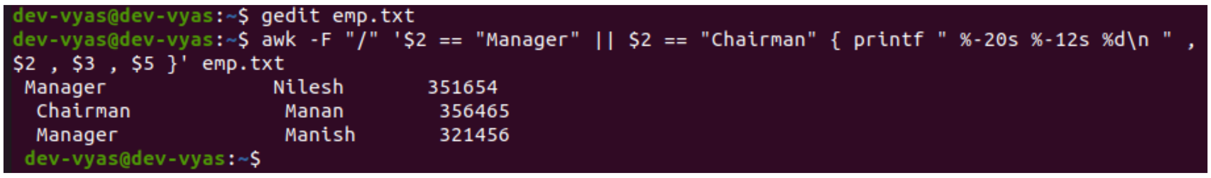
2. For the simple structure $awk ‘Selction_criteria{ action }’ File(s).



```
dev-vyas@dev-vyas:~$ gedit emp.txt
dev-vyas@dev-vyas:~$ awk -f temp.awk emp.txt
001 Salary is more than 45000
002  Salary is more than 45000
003  Salary is less than 45000
004  Salary is more than 45000
005  Salary is more than 45000
```

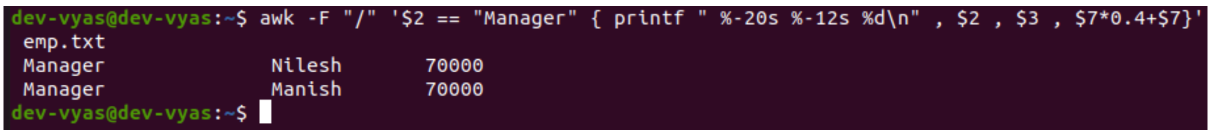
3. Comparison Operators.

001/Manager/Nilesh/Bardoli/351654/9896548954/50000
002/Chairman/Manan/Surat/356465/8944987548/60000
003/ProductManager/Suresh/Navsari/395687/9696985698/55000
004/ServiceManager/Mahesh/Surat/356475/9856898745/50000
005/Analyst/Anand/Ram/369852/9632587455/40000
006/CEO/Delhi/Lakhan/385657/9856321470/120000
007/Manager/Manish/Navsari/321456/9632587410/50000
008/Analyst/Uday/Baroda/325647/98563217581/45000
009/ProductManager/Manju/Ahmedabad/325698/9874563214/55000
010/ServiceManager/Akshay/Tundi/325698/9856325410/50000



```
dev-vyas@dev-vyas:~$ gedit emp.txt
dev-vyas@dev-vyas:~$ awk -F "/" '$2 == "Manager" || $2 == "Chairman" { printf " %-20s %-12s %d\n " ,
$2 , $3 , $5 }' emp.txt
 Manager                Nilesh       351654
  Chairman               Manan        356465
  Manager                Manish       321456
 dev-vyas@dev-vyas:~$
```

4. Number processing and variable use.

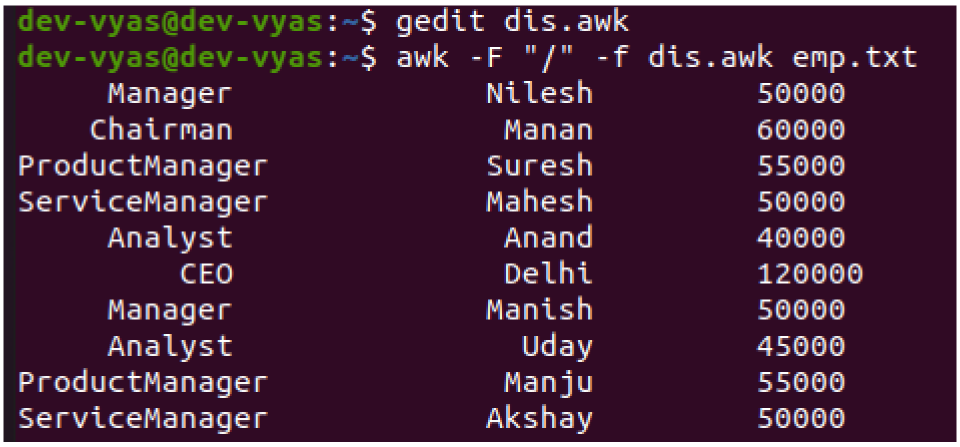


```
dev-vyas@dev-vyas:~$ awk -F "/" '$2 == "Manager" { printf " %-20s %-12s %d\n" , $2 , $3 , $7*0.4+$7}'
 emp.txt
 Manager              Nilesh       70000
 Manager              Manish       70000
dev-vyas@dev-vyas:~$
```

5. Storing AWP program into files and execute through file using option “-f”.

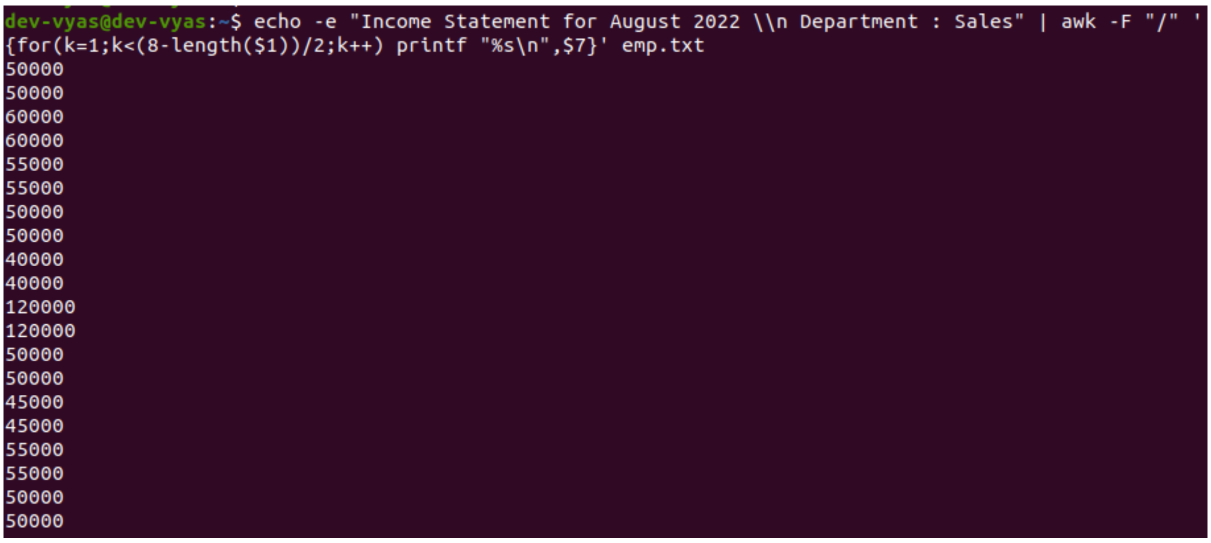
dis.awk

```
{
        NR=3
        NR=7
        {
                printf "%12s\t %15s\t %3d\n",$2,$3,$7
        }
}
```



```
dev-vyas@dev-vyas:~$ gedit dis.awk
dev-vyas@dev-vyas:~$ awk -F "/" -f dis.awk emp.txt
     Manager              Nilesh         50000
    Chairman               Manan         60000
ProductManager            Suresh         55000
ServiceManager            Mahesh         50000
     Analyst               Anand         40000
         CEO               Delhi         120000
     Manager              Manish         50000
     Analyst                Uday         45000
ProductManager             Manju         55000
ServiceManager            Akshay         50000
```

6. Using looping and conditional statements.



```
dev-vyas@dev-vyas:~$ echo -e "Income Statement for August 2022 \\n Department : Sales" | awk -F "/" '
{for(k=1;k<(8-length($1))/2;k++) printf "%s\n",$7}' emp.txt
50000
50000
60000
60000
55000
55000
50000
50000
40000
40000
120000
120000
50000
50000
45000
45000
55000
55000
50000
50000
```

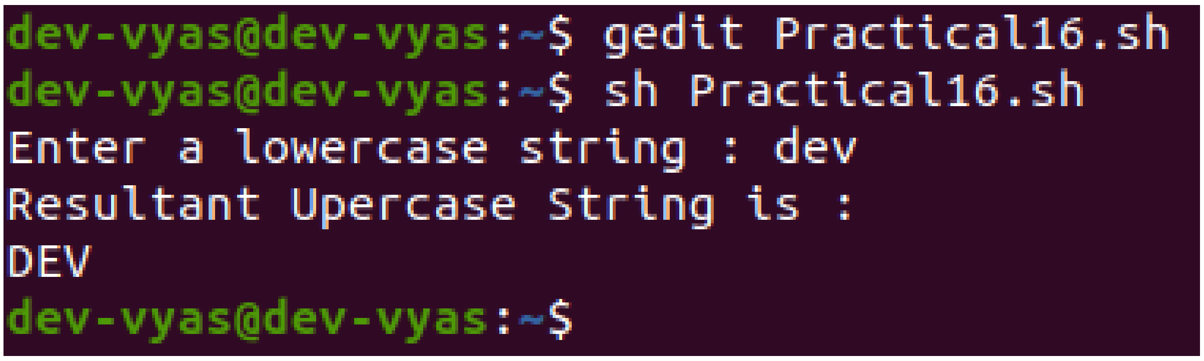
PRACTICAL-16

AIM : Write an awk program using function, which convert each word in a given text into capital.

PROGRAM:

```
read -p "Enter a lowercase string : " string
echo "Resultant Upercase String is : "
echo "$string" | awk '{print toupper($0)}'
```

OUTPUT:



```
dev-vyas@dev-vyas:~$ gedit Practical16.sh
dev-vyas@dev-vyas:~$ sh Practical16.sh
Enter a lowercase string : dev
Resultant Upercase String is :
DEV
dev-vyas@dev-vyas:~$
```

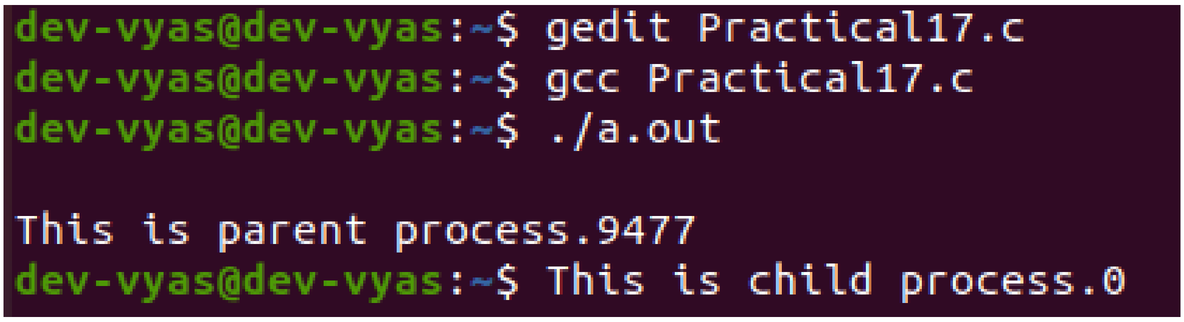
PRACTICAL-17

AIM : Write a program for process creation using C.

PROGRAM:

```
#include<stdio.h>
#include<sys/types.h>
#include<unistd.h>
int main()
{
	int pid;
	pid=fork();
	if(pid==0)
	{
		printf("This is child process.");
		printf("%d",pid);
	}
	else
	{
		printf("\nThis is parent process.");
		printf("%d",pid);
	}
	printf("\n");

}
```

OUTPUT:



```
dev-vyas@dev-vyas:~$ gedit Practical17.c
dev-vyas@dev-vyas:~$ gcc Practical17.c
dev-vyas@dev-vyas:~$ ./a.out

This is parent process.9477
dev-vyas@dev-vyas:~$ This is child process.0
```