

Ex. No: 1.1

BASIC UNIX COMMANDS

Aim

To study and execute Unix commands.

Unix is security conscious, and can be used only by those persons who have an account. *Telnet* (Telephone Network) is a Terminal emulator program for TCP/IP networks that enables users to log on to remote servers.

To *logon*, type **telnet server_ipaddress** in **run** window.

User has to authenticate himself by providing *username* and *password*. Once verified, a greeting and \$ prompt appears. The shell is now ready to receive commands from the user. Options suffixed with a hyphen (-) and arguments are separated by space.

General commands

Command	Function
date	Used to display the current system date and time.
date +%D	Displays date only
date +%T	Displays time only
date +% Y	Displays the year part of date
date +% H	Displays the hour part of time
cal	Calendar of the current month
cal year	Displays calendar for all months of the specified year
cal month year	Displays calendar for the specified month of the year
who	Login details of all users such as their IP, Terminal No, User name,
who am i	Used to display the login details of the user
tty	Used to display the terminal name
uname	Displays the Operating System
uname -r	Shows version number of the OS (kernel).
uname -n	Displays domain name of the server
echo "txt"	Displays the given text on the screen
echo \$HOME	Displays the user's home directory
bc	Basic calculator. Press Ctrl+d to quit
lp file	Allows the user to spool a job along with others in a print queue.
man cmdname	Manual for the given command. Press q to exit
history	To display the commands used by the user since log on.
exit	Exit from a process. If shell is the only process then logs out

Directory commands

Command	Function
pwd	Path of the present working directory
mkdir dir	A directory is created in the given name under the current directory
mkdir dir1 dir2	A number of sub-directories can be created under one stroke
cd subdir	Change Directory. If the <i>subdir</i> starts with / then path starts from root (absolute) otherwise from current working directory.
cd	To switch to the home directory.
cd /	To switch to the root directory.

Command	Function
<code>cd ..</code>	To move back to the parent directory
<code>rmdir subdir</code>	Removes an empty sub-directory.

File commands

Command	Function
<code>cat > filename</code>	To create a file with some contents. To end typing press Ctrl+d . The > symbol means redirecting output to a file. (< for input)
<code>cat filename</code>	Displays the file contents.
<code>cat >> filename</code>	Used to append contents to a file
<code>cp src des</code>	Copy files to given location. If already exists, it will be overwritten
<code>cp -i src des</code>	Warns the user prior to overwriting the destination file
<code>cp -r src des</code>	Copies the entire directory, all its sub-directories and files.
<code>mv old new</code>	To rename an existing file or directory. -i option can also be used
<code>mv f1 f2 f3 dir</code>	To move a group of files to a directory.
<code>mv -v old new</code>	Display name of each file as it is moved.
<code>rm file</code>	Used to delete a file or group of files. -i option can also be used
<code>rm *</code>	To delete all the files in the directory.
<code>rm -r *</code>	Deletes all files and sub-directories
<code>rm -f *</code>	To forcibly remove even write-protected files
<code>ls</code>	Lists all files and subdirectories (blue colored) in sorted manner.
<code>ls name</code>	To check whether a file or directory exists.
<code>ls name*</code>	Short-hand notation to list out filenames of a specific pattern.
<code>ls -a</code>	Lists all files including hidden files (files beginning with .)
<code>ls -x dirname</code>	To have specific listing of a directory.
<code>ls -R</code>	Recursive listing of all files in the subdirectories
<code>ls -l</code>	Long listing showing file access rights (read/write/execute- rw x for user/group/others- ugo).
<code>cmp file1 file2</code>	Used to compare two files. Displays nothing if files are identical.
<code>wc file</code>	It produces a statistics of lines (l), words(w), and characters(c).
<code>chmod perm file</code>	Changes permission for the specified file. (r=4, w=2, x=1) <code>chmod 740 file</code> sets all rights for user, read only for groups and no rights for others

The commands can be combined using the pipeline (|) operator. For example, number of users logged in can be obtained as.

```
who | wc -l
```

Finally to terminate the unix session execute the command **exit** or **logout**.

Result

Thus the study and execution of Unix commands has been completed successfully.

Unix Commands

```
[sec@localhost User1]$ date
Sat Apr  9 13:03:47 IST 2011
[sec@localhost User1]$ date +%D
04/09/11
[sec@localhost User1]$ date +%T
13:05:33
[sec@localhost User1]$ date +%Y
2011
[sec@localhost User1]$ date +%H
13
```

```
[sec@localhost User1]$ cal

    April 2011
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

[sec@localhost User1]$ cal 08 1998
```

```
    August 1998
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
```

```
[sec@localhost User1]$ cal 1800
```

```
1800

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

Unix Commands

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

30

```
[sec@localhost User1]$ who
```

```
root      :0          Apr  9 08:41
User1     pts/0       Apr  9 13:00 (scl-64)
cse1      pts/3       Apr  9 13:18 (scl-41.smkfomra.com)
ece1      pts/4       Apr  9 13:18 (scl-29.smkfomra.com)
```

```
[sec@localhost User1]$ who am i
```

```
User1     pts/0       Apr  9 13:00 (scl-64)
```

```
[sec@localhost User1]$ tty
```

```
/dev/pts/0
```

```
[sec@localhost User1]$ uname
```

```
Linux
```

```
[sec@localhost User1]$ uname -r
```

```
2.4.20-8smp
```

```
[sec@localhost User1]$ uname -n
```

```
localhost.localdomain
```

```
[sec@localhost User1]$ echo "How are you"
```

```
How are you
```

```
[sec@localhost User1]$ echo $HOME
```

```
/home/User1
```

```
[sec@localhost User1]$ echo $USER
```

```
User1
```

```
[sec@localhost User1]$ bc
```

```
bc 1.06
```

```
Copyright 1991-1994, 1997, 1998, 2000 Free Software Foundation,
Inc. 3+5 8 2%3 2
```

```
[sec@localhost loops]$ pwd
```

```
/home/User1/shellscripts/loops
```

```
[sec@localhost User1]$ mkdir filter
```

```
[sec@localhost User1]$ ls
```

```
filter list.sh regexpr shellscripts
```

Unix Commands

```
[sec@localhost User1]$ cd shellscripts/loops/
[sec@localhost loops]$
[sec@localhost loops]$ cd
[sec@localhost User1]$
[sec@localhost loops]$ cd /
[sec@localhost /]$
[sec@localhost /]$ cd /home/User1/shellscripts/loops/
[sec@localhost loops]$ cd ..
[sec@localhost shellscripts]$
[sec@localhost User1]$ rmdir filter
[sec@localhost User1]$ ls
list.sh  regexpr  shellscripts
[sec@localhost User1]$ cat > greet
hi ece-a
wishing u the best
[sec@localhost User1]$ cat greet
hi ece-a
wishing u the best
[sec@localhost User1]$ cat >> greet
bye
[sec@localhost User1]$ cat greet
hi ece-a
wishing u the
best bye
[sec@localhost User1]$ ls
greet  list.sh  regexpr  shellscripts
[sec@localhost User1]$ ls -a
.          .bash_logout  .canna  .gtkrc  regexpr  .viminfo.tmp
..         .bash_profile .emacs  .kde    shellscripts  .xemacs
.bash_history .bashrc      greet   list.sh  .viminfo
[sec@localhost User1]$ ls -l
total 16
-rw-rw-r--  1 User1  User1          32 Apr 11 14:52 greet
-rw-rw-r--  1 User1  User1          30 Apr  4 13:58 list.sh
drwxrwxr-x  2 User1  User1       4096 Apr  9 14:30 regexpr
drwxrwxr-x  7 User1  User1       4096 Apr  4 14:57 shellscripts
[sec@localhost User1]$ cp greet ./regexpr/
[sec@localhost User1]$ ls
```

Unix Commands

```
greet list.sh regexpr shellscripts
[sec@localhost User1]$ ls ./regexpr
demo greet
[sec@localhost User1]$ cp -i greet ./regexpr/
cp: overwrite 'greet'? n
[sec@localhost User1]$ mv greet greet.txt
[sec@localhost User1]$ ls
greet.txt list.sh regexpr shellscripts
[sec@localhost User1]$ mv greet.txt ./regexpr/
[sec@localhost User1]$ ls
list.sh regexpr shellscripts
[sec@localhost User1]$ ls ./regexpr/
demo greet.txt
[sec@localhost User1]$ ls
fact.sh list.sh prime.sh regexpr shellscripts
[sec@localhost User1]$ rm -i *.sh
rm: remove regular file `fact.sh'? y
rm: remove regular file `list.sh'? n
rm: remove regular file `prime.sh'? y
[sec@localhost User1]$ ls
list.sh regexpr shellscripts
[sec@localhost User1]$ wc list.sh
      4      9     30 list.sh
[sec@localhost User1]$ wc -l list.sh
      4 list.sh
[sec@localhost User1]$ cmp list.sh fact.sh
list.sh fact.sh differ: byte 1, line 1
[sec@localhost User1]$ ls -l list.sh
-rw-rw-r-- 1 User1 User1      30 Apr  4 13:58 list.sh
[sec@localhost User1]$ chmod ug+x list.sh
[sec@localhost User1]$ ls -l list.sh
-rwxrwxr-- 1 User1 User1      30 Apr  4 13:58 list.sh
[sec@localhost User1]$ chmod 740 list.sh
[sec@localhost User1]$ ls -l list.sh
-rwxr----- 1 User1 User1      30 Apr  4 13:58 list.sh
```

Ex. No: 1.2**Simple Filters****Aim**

To query a data file using filter commands in unix.

Filters are the central commands of the UNIX tool kit. It acts on data file where lines are *records*, *fields* delimited by a character not used by the data (mostly |, default is white space). The output is a set of records and the input file is unaltered by these commands.

20057801	Aarthi	ECE	CTS	36000
20057702	Albert Jerry	CSE	Wipro	25000
20057903	Arun	IT	Ramco	12000
20057904	Diwakar	IT	TCS	10500
20057705	Geetha	CSE	Infosys	23000
20057806	Irudayaraj	ECE	Polaris	30000
20057707	Jaya Prakash	CSE	Ramco	28000
20058008	Mahesh	EEE	Microsoft	5000
20057909	Manimaran	IT	Microsoft	9000
20058010	Mohammed Mukthar	EEE	Oracle	6000
20057711	Prithivi Rajan	CSE	Ramco	25000
20057712	Pushpak Chander	CSE	CTS	27500
20057713	Ramesh	CSE	Wipro	24000
20057817	Smitha	ECE	Ramco	30000

stud file

Command	Function
head used to display the first few records (10 records by default)	
head stud	Displays first 10 records by default
head -5 stud	Displays first 5 records
head -1 stud wc -c	length of first record
tail used to display the last few records (10 records by default)	
tail stud	Displays last 10 records by default
tail -5 stud tee last5	Last 5 records listed & stored in file <i>last5</i> using <i>tee</i>
cut used to extract specific fields. The d option specifies the delimiter and f for specifying the field list. The c option may be used if extraction is done character wise	
cut -d \ -f 1,3,4 stud	Fields 1,3,4 listed
cut -d \ -f 2-4 stud	Fields 2,3,4 listed
paste -d \ list1 list2	merges two cut files <i>list1</i> and <i>list2</i>
sort reorders the file as per ASCII sequence. The t option is used to specify delimiter	
sort stud	Sorted on 1 st column by default
sort -t \ +2 stud	Sort as per 3 rd column
sort -c stud	Check if file is sorted using c option
sort -t \ +3 -4 +4 stud	Sorting on secondary keys
sort -t \ -nr +4 stud	Sort on numeric field using n option, r for reverse
uniq stud	Display unique entries in a sorted file
nl display file content with lines numbered. The s option is used to specify separator	
nl -s " " stud	Displays entries numbered with separator
tr translates characters. Can be used to change text case. It works with standard input <	
tr '[a-z]' '[A-Z]' < stud	Changes text to upper case

Result

Thus information retrieval using filters has been completed successfully.

Unix Commands

```
[sec@localhost filters]$ head stud
20057801|Aarthi          |ECE |CTS          |36000
20057702|Albert Jerry    |CSE |Wipro          |25000
20057903|Arun                |IT  |Ramco          |12000
20057904|Diwakar             |IT  |TCS            |10500
20057705|Geetha              |CSE |Infosys        |23000
20057806|Irudayaraj          |ECE |Polaris        |30000
20057707|Jaya Prakash        |CSE |Ramco          |28000
20058008|Mahesh              |EEE |Microsoft      |5000
20057909|Manimaran           |IT  |Microsoft      |9000
20058010|Mohammed Mukthar    |EEE |Oracle          |6000
[sec@localhost filters]$ head -4 stud
20057801|Aarthi          |ECE |CTS          |36000
20057702|Albert Jerry    |CSE |Wipro          |25000
20057903|Arun                |IT  |Ramco          |12000
20057904|Diwakar             |IT  |TCS            |10500
[sec@localhost filters]$ head -1 stud | wc -c
49
[sec@localhost filters]$ tail stud
20058008|Mahesh          |EEE |Microsoft      |5000
20057909|Manimaran           |IT  |Microsoft      |9000
20058010|Mohammed Mukthar |EEE |Oracle          |6000
20057711|Prithivi Rajan     |CSE |Ramco          |25000
20057712|Pushpak Chander    |CSE |CTS            |27500
20057713|Ramesh          |CSE |Wipro          |24000
20057817|Smitha             |ECE |Ramco          |30000
20057718|Sri Gurumoorthy    |IT  |Microsoft      |11000
20057719|Tamil Selvi        |EEE |CTS            |3500
20057720|Thamotharan        |IT  |CTS            |9000
[sec@localhost filters]$ tail -2 stud | tee last2
20057719|Tamil Selvi        |EEE |CTS            |3500
20057720|Thamotharan        |IT  |CTS            |9000
[sec@localhost filters]$ cat last2
20057719|Tamil Selvi        |EEE |CTS            |3500
20057720|Thamotharan        |IT  |CTS            |9000
[sec@localhost filters]$ cut -d \| -f 2,4-5 stud
20057801|Aarthi          |ECE
20057702|Albert Jerry    |CSE
20057903|Arun                |IT
20057904|Diwakar             |IT
20057705|Geetha              |CSE
20057806|Irudayaraj          |ECE
20057707|Jaya Prakash        |CSE
20058008|Mahesh              |EEE
```


Unix Commands

```
20057909|Manimaran      |IT
20058010|Mohammed Mukthar |EEE
20057711|Prithivi Rajan   |CSE
20057712|Pushpak Chander  |CSE
20057713|Ramesh          |CSE
20057817|Smitha             |ECE
20057718|Sri Gurumoorthy   |IT
20057719|Tamil Selvi       |EEE
20057720|Thamotharan       |IT
```

```
[sec@localhost filters]$ cut -d \| -f 2,4 stud > nameorg
```

```
[sec@localhost filters]$ cut -d \| -f 5 stud > sal
```

```
[sec@localhost filters]$ paste -d \| nameorg sal
```

```
Aarthi          |CTS          |36000
Albert Jerry    |Wipro        |25000
Arun            |Ramco        |12000
Diwakar         |TCS          |10500
Geetha          |Infosys      |23000
Irudayaraj     |Polaris      |30000
Jaya Prakash    |Ramco        |28000
Mahesh          |Microsoft    |5000
Manimaran       |Microsoft    |9000
Mohammed Mukthar|Oracle       |6000
Prithivi Rajan  |Ramco        |25000
Pushpak Chander |CTS          |27500
Ramesh          |Wipro        |24000
Smitha          |Ramco        |30000
Sri Gurumoorthy |Microsoft    |11000
Tamil Selvi     |CTS          |3500
Thamotharan     |CTS          |9000
```

```
[sec@localhost filters]$ sort stud
```

```
20057702|Albert Jerry    |CSE |Wipro        |25000
20057705|Geetha          |CSE |Infosys      |23000
20057707|Jaya Prakash    |CSE |Ramco        |28000
20057711|Prithivi Rajan  |CSE |Ramco        |25000
20057712|Pushpak Chander |CSE |CTS          |27500
20057713|Ramesh          |CSE |Wipro        |24000
20057718|Sri Gurumoorthy |IT  |Microsoft    |11000
20057719|Tamil Selvi     |EEE |CTS          |3500
20057720|Thamotharan     |IT  |CTS          |9000
20057801|Aarthi          |ECE |CTS          |36000
20057806|Irudayaraj     |ECE |Polaris      |30000
20057817|Smitha          |ECE |Ramco        |30000
20057903|Arun            |IT  |Ramco        |12000
20057904|Diwakar         |IT  |TCS          |10500
20057909|Manimaran       |IT  |Microsoft    |9000
20058008|Mahesh          |EEE |Microsoft    |5000
20058010|Mohammed Mukthar|EEE |Oracle       |6000
```

Unix Commands

```
[sec@localhost filters]$ sort -t \| +1 stud
```

20057801 Aarthi	ECE	CTS	36000
20057702 Albert Jerry	CSE	Wipro	25000
20057903 Arun	IT	Ramco	12000
20057904 Diwakar	IT	TCS	10500
20057705 Geetha	CSE	Infosys	23000
20057806 Irudayaraj	ECE	Polaris	30000
20057707 Jaya Prakash	CSE	Ramco	28000
20058008 Mahesh	EEE	Microsoft	5000
20057909 Manimaran	IT	Microsoft	9000
20058010 Mohammed Mukthar	EEE	Oracle	6000
20057711 Prithivi Rajan	CSE	Ramco	25000
20057712 Pushpak Chander	CSE	CTS	27500
20057713 Ramesh	CSE	Wipro	24000
20057817 Smitha	ECE	Ramco	30000
20057718 Sri Gurumoorthy	IT	Microsoft	11000
20057719 Tamil Selvi	EEE	CTS	3500
20057720 Thamotharan	IT	CTS	9000

```
[sec@localhost filters]$ sort -t \| +3 -4 +2 stud
```

20057712 Pushpak Chander	CSE	CTS	27500
20057801 Aarthi	ECE	CTS	36000
20057719 Tamil Selvi	EEE	CTS	3500
20057720 Thamotharan	IT	CTS	9000
20057705 Geetha	CSE	Infosys	23000
20058008 Mahesh	EEE	Microsoft	5000
20057718 Sri Gurumoorthy	IT	Microsoft	11000
20057909 Manimaran	IT	Microsoft	9000
20058010 Mohammed Mukthar	EEE	Oracle	6000
20057806 Irudayaraj	ECE	Polaris	30000
20057711 Prithivi Rajan	CSE	Ramco	25000
20057707 Jaya Prakash	CSE	Ramco	28000
20057817 Smitha	ECE	Ramco	30000
20057903 Arun	IT	Ramco	12000
20057904 Diwakar	IT	TCS	10500
20057713 Ramesh	CSE	Wipro	24000
20057702 Albert Jerry	CSE	Wipro	25000

```
[sec@localhost filters]$ sort -t \| -nr +4 stud
```

20057801 Aarthi	ECE	CTS	36000
20057817 Smitha	ECE	Ramco	30000
20057806 Irudayaraj	ECE	Polaris	30000
20057707 Jaya Prakash	CSE	Ramco	28000
20057712 Pushpak Chander	CSE	CTS	27500
20057711 Prithivi Rajan	CSE	Ramco	25000
20057702 Albert Jerry	CSE	Wipro	25000
20057713 Ramesh	CSE	Wipro	24000
20057705 Geetha	CSE	Infosys	23000

Unix Commands

```
20057903|Arun          |IT   |Ramco          |12000
20057718|Sri Gurumoorthy    |IT   |Microsoft      |11000
20057904|Diwakar            |IT   |TCS            |10500
20057909|Manimaran          |IT   |Microsoft      |9000
20057720|Thamotharan        |IT   |CTS            |9000
20058010|Mohammed Mukthar   |EEE  |Oracle         |6000
20058008|Mahesh             |EEE  |Microsoft      |5000
20057719|Tamil Selvi        |EEE  |CTS            |3500
```

```
[sec@localhost filters]$ tr ' [a-z]' ' [A-Z]' < stud
```

```
20057801|AARTHI          |ECE  |CTS            |36000
20057702|ALBERT JERRY       |CSE  |WIPRO          |25000
20057903|ARUN               |IT   |RAMCO          |12000
20057904|DIWAKAR            |IT   |TCS            |10500
20057705|GEETHA             |CSE  |INFOSYS        |23000
20057806|IRUDAYARAJ        |ECE  |POLARIS        |30000
20057707|JAYA PRAKASH       |CSE  |RAMCO          |28000
20058008|MAHESH             |EEE  |MICROSOFT      |5000
20057909|MANIMARAN          |IT   |MICROSOFT      |9000
20058010|MOHAMMED MUKTHAR   |EEE  |ORACLE         |6000
20057711|PRITHIVI RAJAN     |CSE  |RAMCO          |25000
20057712|PUSHPAK CHANDER    |CSE  |CTS            |27500
20057713|RAMESH             |CSE  |WIPRO          |24000
20057817|SMITHA            |ECE  |RAMCO          |30000
20057718|SRI GURUMOORTHY    |IT   |MICROSOFT      |11000
20057719|TAMIL SELVI        |EEE  |CTS            |3500
20057720|THAMOTHARAN        |IT   |CTS            |9000
```

```
[sec@localhost filters]$ nl -s "|" stud
```

```
1|20057801|Aarthi          |ECE  |CTS            |36000
2|20057702|Albert Jerry       |CSE  |Wipro          |25000
3|20057903|Arun               |IT   |Ramco          |12000
4|20057904|Diwakar            |IT   |TCS            |10500
5|20057705|Geetha             |CSE  |Infosys        |23000
6|20057806|Irudayaraj        |ECE  |Polaris        |30000
7|20057707|Jaya Prakash       |CSE  |Ramco          |28000
8|20058008|Mahesh             |EEE  |Microsoft      |5000
9|20057909|Manimaran          |IT   |Microsoft      |9000
10|20058010|Mohammed Mukthar   |EEE  |Oracle         |6000
11|20057711|Prithivi Rajan     |CSE  |Ramco          |25000
12|20057712|Pushpak Chander    |CSE  |CTS            |27500
13|20057713|Ramesh             |CSE  |Wipro          |24000
14|20057817|Smitha            |ECE  |Ramco          |30000
15|20057718|Sri Gurumoorthy    |IT   |Microsoft      |11000
16|20057719|Tamil Selvi        |EEE  |CTS            |3500
17|20057720|Thamotharan        |IT   |CTS            |9000
```

Ex. No: 1.3

Regular Expression

Aim

To search for regular expression in a file using grep command in unix.

A frequent requirement is to look for a pattern or expression in a file. Unix handles this feature through **grep** and **egrep**. **grep** uses an regular expression to display lines that match and **egrep** enables searching for multiple patterns. Its usage is `grep options searchtext filename`

```
THIS LINE IS THE 1ST UPPER CASE LINE IN THIS FILE.
this line is the 1st lower case line in this file.
This Line Has All Its First Character Of The Word With Upper Case.

Two lines above this line is
empty. vim Word Navigation

You may want to do several navigation in relation to words, such as:

1. e - go to the end of the current word.
2. E - go to the end of the current WORD.
3. b - go to the previous word.
4. B - go to the previous WORD.

WORD - WORD consists of a sequence of non-blank characters
Word - word consists of a sequence of letters, digits and
underscores. telnet 172.16.4.256
```

demo file

Command	Function
grep this demo	Lists the lines that contains the string <i>this</i>
grep 'end of' demo	Quotes mandatory for text containing space
grep this demo*	Search <i>this</i> in multiple files
grep -c to demo	Number of occurrence of the word <i>to</i> in the file
grep -n sequence demo	Display line numbers along with matching lines
grep -v word demo	Displays lines that does not contain the text <i>word</i>
grep -l vim *	Displays files containing text <i>vim</i>
grep -i WORD demo	Search for text ignoring case differences
grep '^ [0-9]' demo	Lines that start with a number
grep '[0-9]\$' demo	Lines that end with a number
ls -l grep "^d"	Display the subdirectory names
grep -c "^\$" demo	Display count of blank lines in the file.
grep "2....\$" stud	Display lines that ends in the range 20000–29999
egrep "lower UPPER" demo	Display lines that match either <i>lower</i> or <i>upper</i>
egrep "(previous current) word" demo	Display lines that match either <i>previous word</i> or <i>current word</i>

Result

Thus searching text patterns in files using grep has been completed successfully.

Unix Commands

```
[sec@localhost regexpr]$ grep this demo
this line is the 1st lower case line in this
file. Two lines above this line is empty.
[sec@localhost regexpr]$ grep 'end of' demo
1. e - go to the end of the current word.
2. E - go to the end of the current WORD.
[sec@localhost regexpr]$ grep -c to demo
5
[sec@localhost regexpr]$ grep -n sequence demo
15:WORD - WORD consists of a sequence of non-blank characters
16:Word - word consists of a sequence of letters, digits and underscores.
[sec@localhost regexpr]$ grep -v word demo
THIS LINE IS THE 1ST UPPER CASE LINE IN THIS FILE.
this line is the 1st lower case line in this file.
This Line Has All Its First Character Of The Word With Upper Case.
Two lines above this line is
empty. vim Word Navigation

2. E - go to the end of the current
WORD. 4. B - go to the previous WORD.
WORD - WORD consists of a sequence of non-blank characters

telnet 172.16.4.256

[sec@localhost regexpr]$ grep -l vim *
demo readme
[sec@localhost regexpr]$ grep -i WORD demo
This Line Has All Its First Character Of The Word With Upper
Case. vim Word Navigation
You may want to do several navigation in relation to words, such as:
1. e - go to the end of the current word.
2. E - go to the end of the current WORD.
3. b - go to the previous word.
4. B - go to the previous WORD.
WORD - WORD consists of a sequence of non-blank characters
Word - word consists of a sequence of letters, digits and underscores.

[sec@localhost regexpr]$ grep '^[0-9]' demo
1. e - go to the end of the current word.
2. E - go to the end of the current WORD.
3. b - go to the previous word.
4. B - go to the previous WORD.
```

Unix Commands

```
[sec@localhost regexpr]$ grep '[0-9]' demo
telnet 172.16.4.256
[sec@localhost User1]$ ls -l | grep "^d"
drwxrwxr-x    2 User1    User1          4096 Apr  9 14:30 regexpr
drwxrwxr-x    7 User1    User1          4096 Apr  4 14:57 shellscripts
[sec@localhost regexpr]$ grep -c "^$" demo
5
[sec@localhost regexpr]$ egrep "lower|UPPER" demo
THIS LINE IS THE 1ST UPPER CASE LINE IN THIS FILE.
this line is the 1st lower case line in this file.
[sec@localhost regexpr]$ egrep "(previous|current) word" demo
1. e - go to the end of the current
word. 3. b - go to the previous word.
```