Chapter Table of Contents

Chapter 4.2

Advanced Features of Vi Editor

Aim

To understand features of vi editor

Instructional Objectives

After completing this chapter, you should be able to:

- Demonstrate how to execute a shell command using macros
- Illustrate how to set various window properties with vi editor
- Demonstrate how to auto indent lines and insert line numbers for the file
- Discuss the basic elements of communication process
- Demonstrate how UNIX administrators communicate with other users using mail, wall, send, mesg, and ftp

Learning Outcomes

At the end of this chapter, you are expected to:

- Explain the procedure to run macros in shell command
- List the commands used to set widow position
- Write the commands used to auto indent and display line numbers
- Identify the basic elements of communication process
- Use various commands to communicate with other users

4.2.1 Running Shell Command Macros

You can record macros in Vi and run them later when required. Macros are used when you want to execute same set of commands within a file.

The steps to record macro and run in vi editor are as follows:

1. Press q to record a macro followed by a lowercase character which represent the name of macro.
2. Perform the action that you want to record and repeat
3. Press q to stop recording
4. You can run the macro by pressing @ followed by macro name
5. If you want to run the macro multiple times the press NN@ macro name, NN is number of times you want to repeat the macro.

Let us see an example to record and run a macro in vi editor:

Type the following command to open a file " copy-test1.txt"

$ vi copy-test1.txt

When the file is open, go to insert mode (Type ESC i) and type "hello" in file as shown in following screenshot:

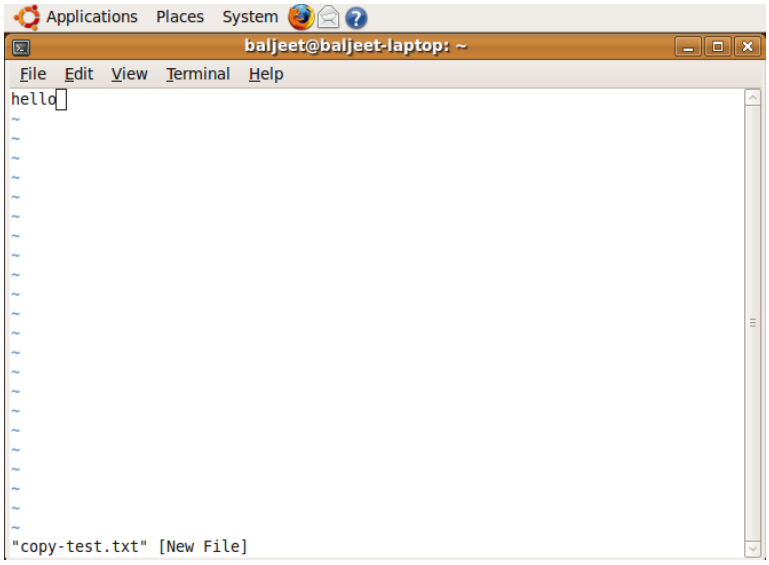

Figure 4.2.1

Now, Type Esc q followed by b for starting the macro recording and store it in register b. as soon as this action is taken, it will display a message as “recording” at the bottom of vi editor as shown in screenshot below:

Here,

q-indicate start of recording

b- indicate name of register where recording is saved.

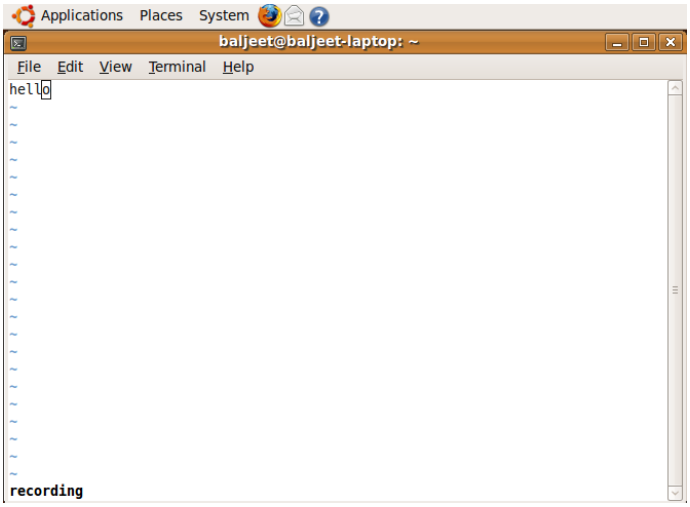

Figure 4.2.2

In this file we want to copy the contents of first line to line 2 and repeat it for 6 times. To store the necessary action in macro, type Esc yy followed by p as shown in following screenshot:

yy is used to copy the current line and p is used to paste the line that was just copied.

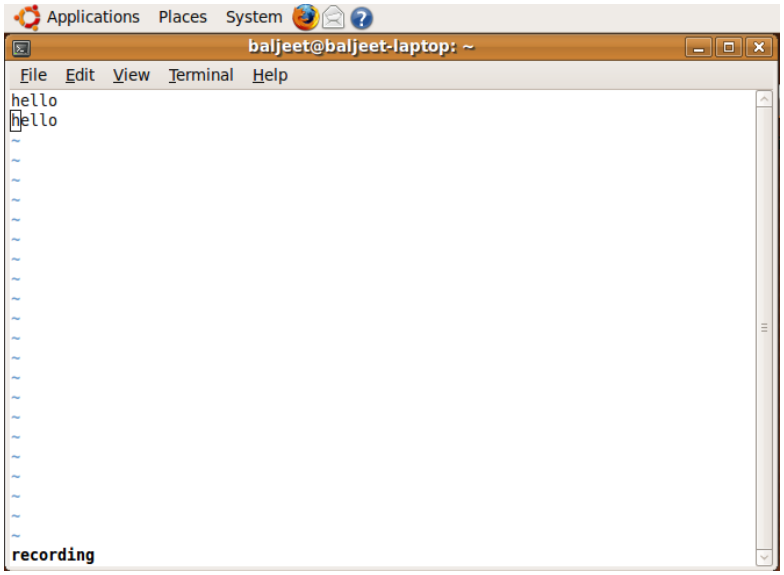

Figure 4.2.3

While recording a macro, you can perform some more actions which will also be recorded. Once recording is done, you can press q and the recording will be stopped.

Now, we will press q to stop recording as shown in the screenshot below:

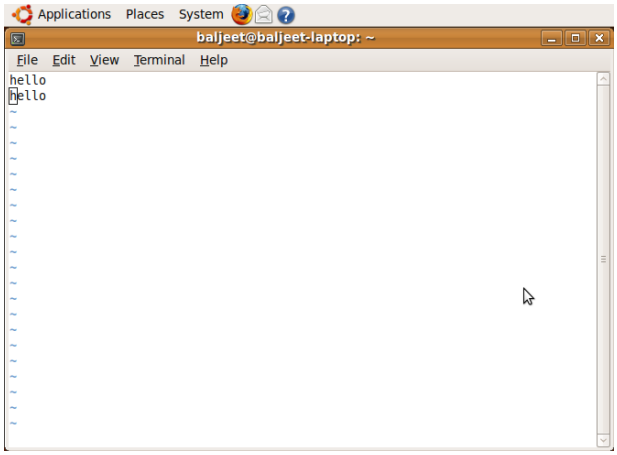

Figure 4.2.4

Now the macro is saved and our actions are recorded in macro. As we want to repeat the action 6 times, you can type: 6@b.

Here,

B is the name of macro.

@b represents execute the macro “b” one time

6@b will rereat the macro “b” 6 time by just coping “hello” to next line for 6 times as shown in following screen:

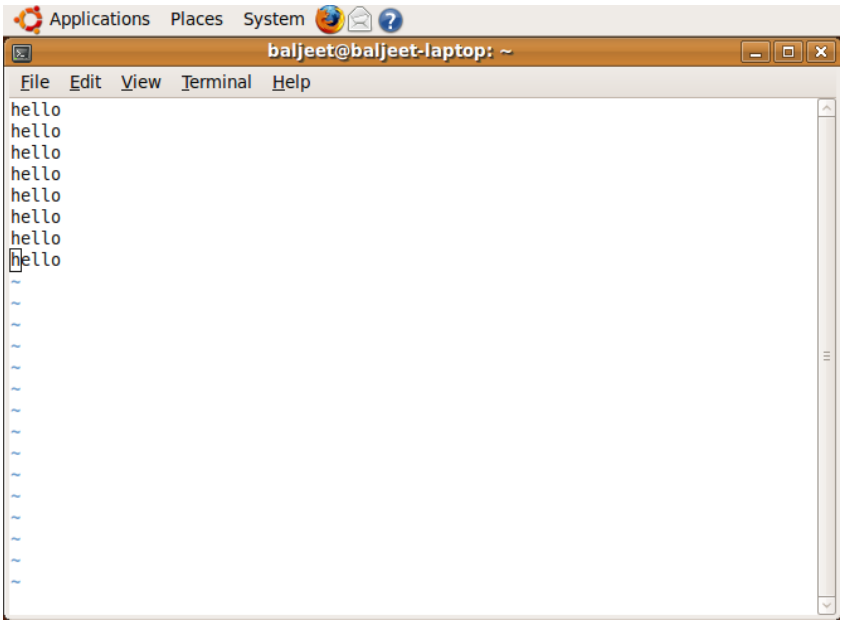

Figure 4.2.5

The following table shows the command macros and action performed by them:

:!cmd	Execute shell command cmd; you can add these special characters to indicate:% name of current file # name of last file edited.
!!cmd	Executes shell command cmd, places output in file starting at current line
:!!	Execute last shell command.
:r!cmd	Reads and inserts output from cmd
:f New_file_Name	Rename current file to New_file_Name

:w !cmd	Sends currently edited file to cmd as standard input and execute cmd
:cd new_dir	Changes current working directory to new_dir
:sh	Starts a sub-shell(ctrl-d returns to editor)
:so file	Reads and executes commands in file(file is a shell script)

Self-assessment Questions

1) Macros are used when you want to execute different set of commands within a file
   a) True b) False

2) You can run a macro by pressing _____followed by macro name
   a) # b) $
   c) @ d) &

3) _____ Executes shell command cmd, places output in file starting at current line
   a) :!cmd b) !!cmd
   c) :cd new_dir d) :so files

4) _____ Changes current working directory to new_dir
   a) :!cmd b) !!cmd
   c) :cd new_dir d) :so files

5) _____ Reads and executes commands in file(file is a shell script)line
   a) :!cmd b) !!cmd
   c) :cd new_dir d) :so files

4.2.2 Set Window

When you want to set the number of lines in a text window, :set window command will be used.

:set window=n

4.2.3 Set Auto Indent

Indentations add readability in the code. The vi editor has a facility to provide this indention automatically. The set command can be used for auto indentation, [enter] key in input mode, the cursor is placed in next line at the current indentation. This auto indentation can be enabled as well as disabled. These commands can be used in command mode only. So if you are in the insert mode currently then press ESC to come in the command mode and enter the command. The **:set ai** command is used to set auto indentation. To disable auto indentation type below command **:set noai**

For example:
Say you type following lines in vi editor without using auto indentation –
Hi How are you?
I am fine Thankyou.

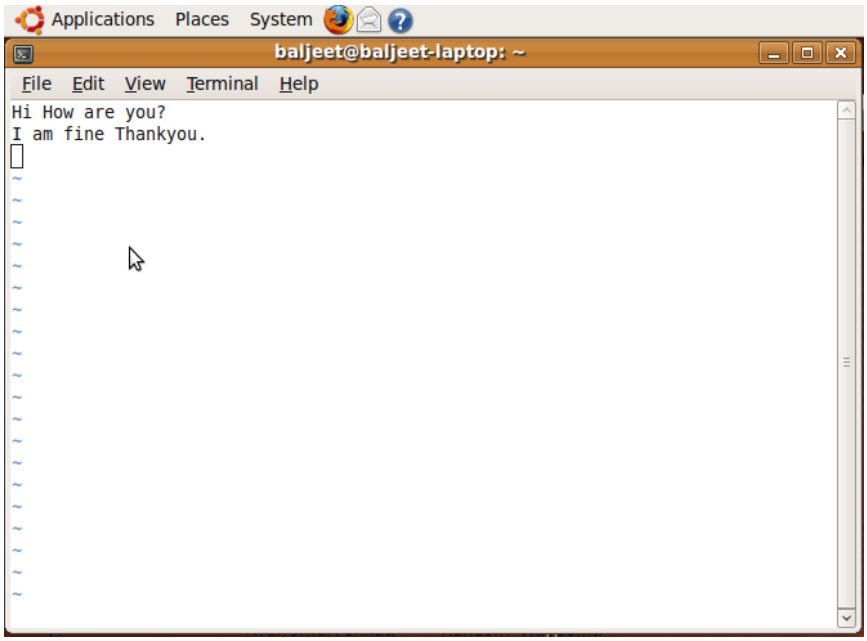

Figure 4.2.6

Following is the screenshot with auto indentation :set ai

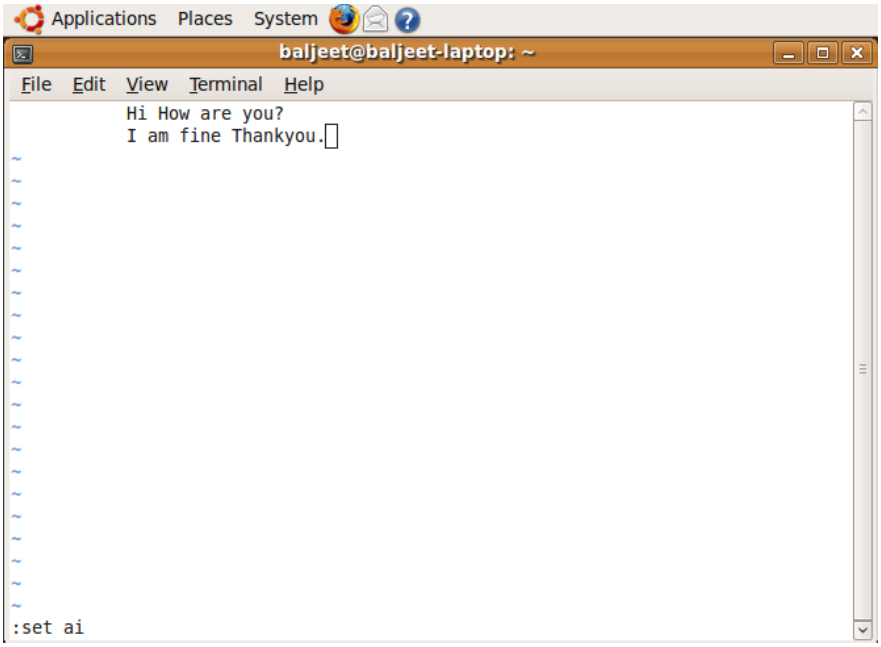

Figure 4.2.7

Following is the screenshot after using disabling auto indentation :set noai

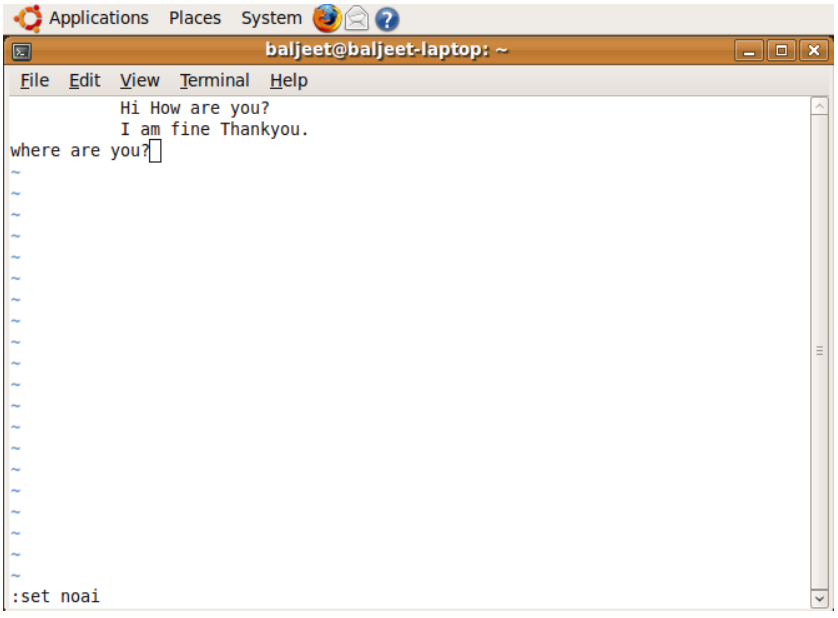

Figure 4.2.8

4.2.4 Set Number

If a file is having line numbers, it becomes easy to debug or trace. We can add the line numbers to a file by using a command. These line numbers can be enabled or disabled as required.

Under command mode, type set number as shown below:

To enable

:set number

Or

:set nu

To disable

:set nonumber

Or

:set nonu

Following is the screenshot to enable auto number:

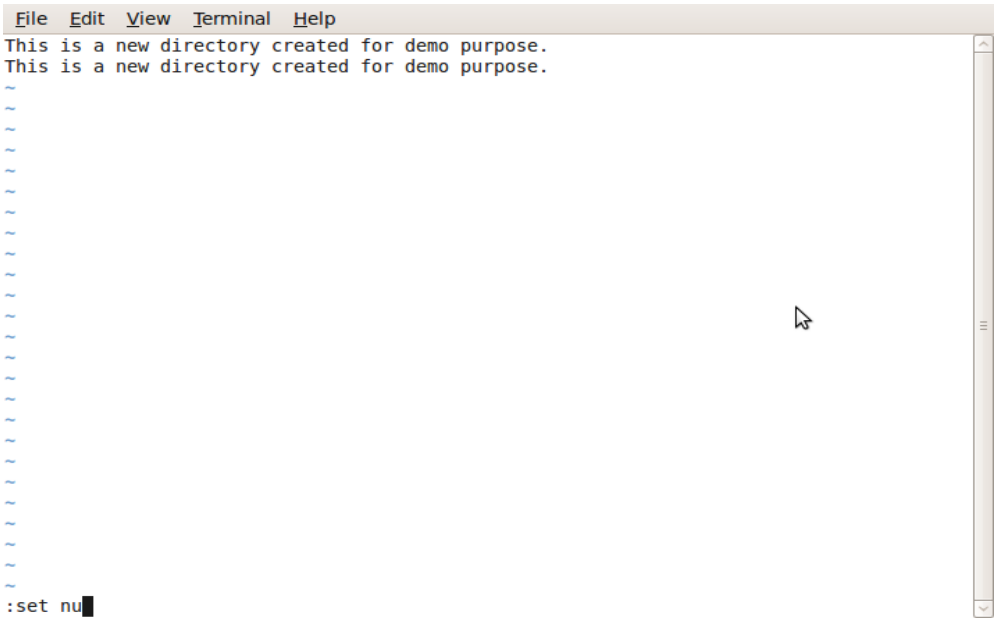

Figure 4.2.9

Following is the screen when auto number is enabled:

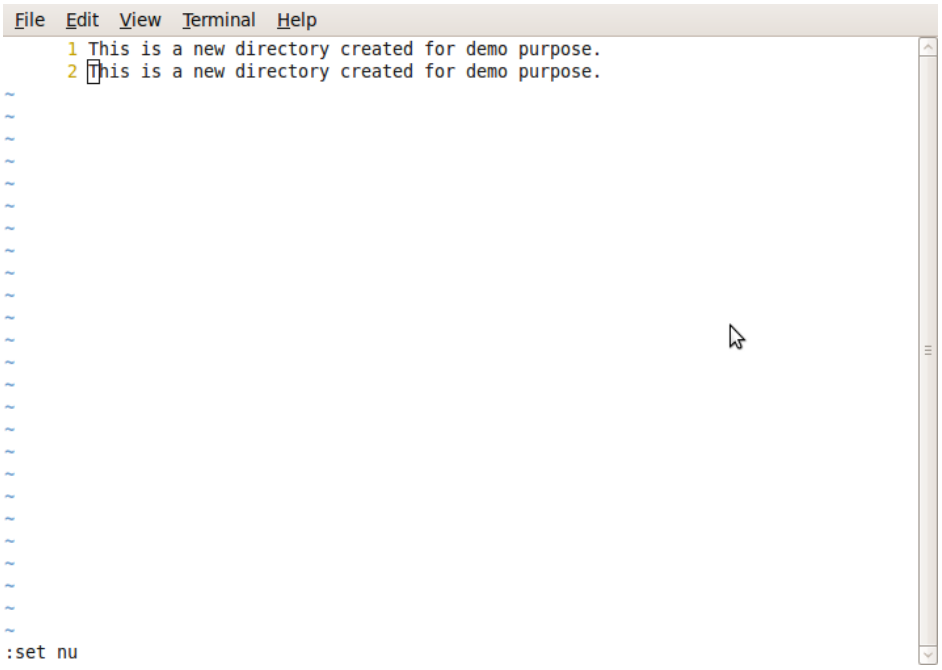

Figure 4.2.10

Following is the screenshot when auto number is disabled using :set nonu

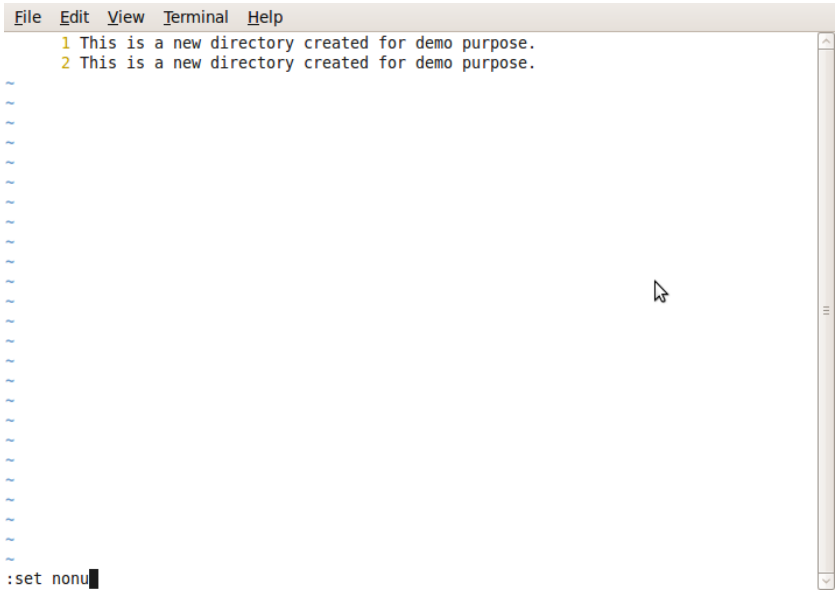

Figure 4.2.11

Following screenshot is result of :set nonu

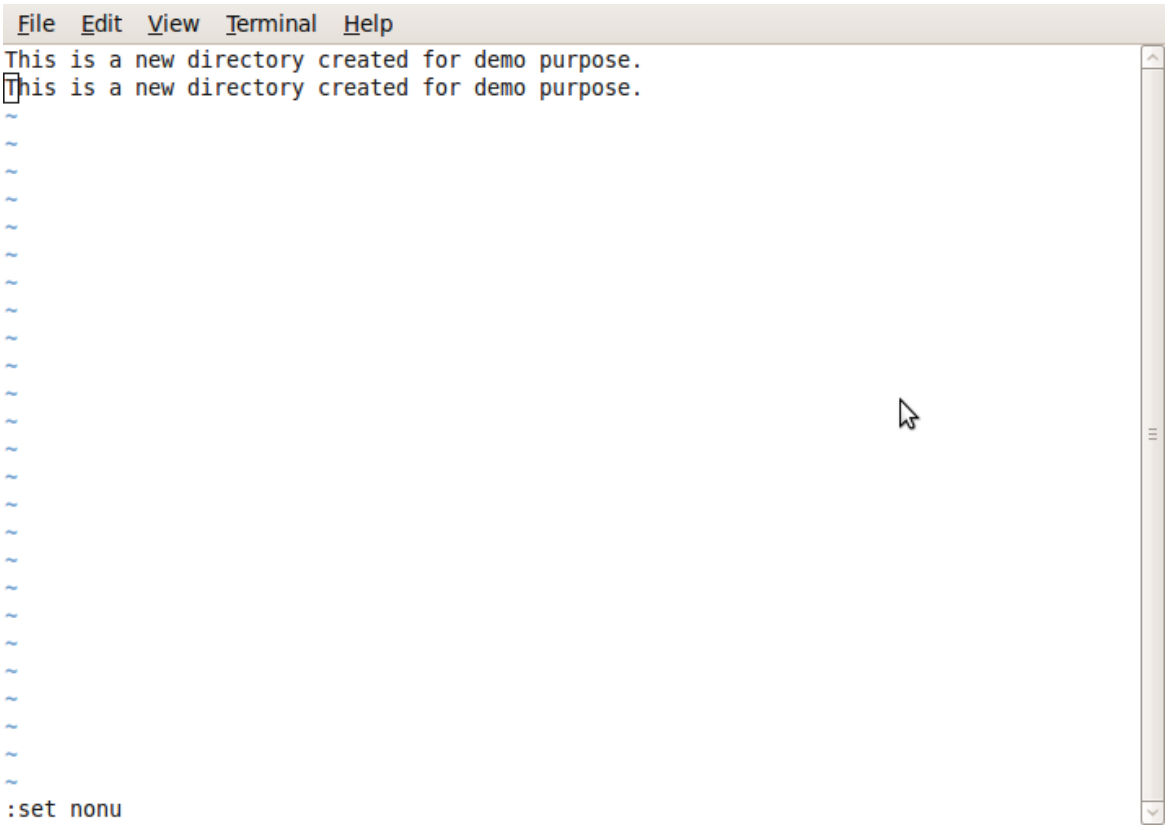

Figure 4.2.12

Self-assessment Questions

6) Set window,_____
   a) Sets number of lines in a text window
   b) Sets number of words in a text window
   c) Sets number of characters in a text window
   d) None of these

7) _____ is used to disable auto indentation
   a) :set ai b) :set noai
   c) :set aill d) :set aion

8) _____ is used to enable line numbers in the vi editor
   a: set ai b) :set noai
   c) :set aill d) :set aion

9) _____ is used to enable auto indentation
   a: set ai b) :set noai
   c) :set aill d) :set aion

4.2.5 Communicating with Other Users

a) Mail

Using mail we can easily send messages to users on network if they are not even logged in. This is different from **write** command as in case of write user must be logged in. the email can be sent to single user as well as to multiple users. The command used for sending email to single user is $mail user_name. if you want to send email to multiple users then we can have a space separated list of the users after $mail as $mail user1 user2 user3.

e.g. To send mail to single user

$mail user_name2

Subject: Hi

Message to be sent

Ctrl+d

Or

To send mail to multiple user

$mail user2 user5 user6

Subject: Broadcast message for data save

System is going to be Shutdown at 6:00 PM.

Please take backup of your data.

Ctrl+d

b) wall

This command can be used by super user to write to any user on the network
It can be executed as shown below:
#/etc/wall
System is going down in next 5 minutes
Please save your data.
ctrl+d
All users who are logged in will see above message on their terminals.
Note: Command prompt in case of super user would be # instead of $.

c) send

Write command or Send
The write command can be used by any user to send message on someone else's terminal, provided the recipient of the message permits communication.
$write user_name2
Hello, How are you?
Ctrl+d

d) mesg,

The message would reach to the user_name2 if he logged on and given permission to receive the message using mesg command.
mesg –y
if you do not want to be disturbed, you can block messages from other users through following command:
mesg -n
Note: A super user can write to any terminal, irrespective of mesg set as –y or –n.

e) ftp

FTP stands for File Transfer Protocol.

ftp command is used to transfer files between hosts.

$ftp host2

It prompts username and password for host2

After successful authentication, **ftp>** prompt would appear to further execution of ftp internal commands.

We can get detail of ftp internal commands list by typing help, as shown below:

$ftp>help

File and Directory handling

We can execute all the basic unix commands at ftp prompt on remote machine like – **pwd, ls, cd, mkdir and chmod.**

Similarly, **delete**, and **rename** can also be used.

To execute similar commands on the local machine, we need to use **!** along with the command

As shown below:

ftp>pwd

It shows working directory path on the remote machine.

ftp>!pwd

It shows working directory path on the local machine.

Transferring files

put and mput

put upload or sends a single file to the remote machine

mput upload multiple files to remote machine.

get and mget

Similarly get and mget used to copy files from remote machine to local machine.

Self-assessment Questions

10) Using mail we can easily send messages to users on network if they are not even logged in

a) TRUE b) FALSE

11) This command can be used by super user to write to any user on the network

a) Mail b) Wall

c) Send d) Mesg

12) This command can be used by any user to send message on someone else's terminal, provided the recipient of the message permits communication

a) Mail b) Wall

c) Send d) Mesg

13) This command can be used by user to send message to other user if he is logged on

a) Mail b) Wall

c) Send d) Mesg

Summary

- You can record macros in Vi and run them later when required. Macros are used when you want to execute same set of commands within a file.
- Sometimes a user wants to customize the vi environment according to their requirement. For example, you would like to define some keys for a sequence of commands which are used frequently. The set commands allow you to do so.
- Different variables in vi environment are controlled by :set command. Set command is used with a variable name. Let us study some of the variables controlled by :set command.
- Set Window: Sets number of lines in a text window.
- Sometimes programmers need to provide indentation in their code for better readability. When this set command for auto indentation is used, [enter] key in input mode, the cursor is placed in next line at the current indentation.
- It is easy to debug or trace a file if the lines in the program code are numbered duly. Set number is used in vi editor to enable line numbers in the vi editor.
- Using mail we can easily send messages to users on network if they are not even logged in. This is different from write command as in case of write user must be logged in.
- Wall command can be used by super user to write to any user on the network
- Write command or Send command can be used by any user to send message on someone else's terminal, provided the recipient of the message permits communication.
- The message would reach to the user_name2 if he logged on and given permission to receive the message using mesg command.
- FTP stands for File Transfer Protocol. ftp command is used to transfer files between hosts.

Terminal Questions

1. Demonstrate how to execute a shell command using macros.
2. Illustrate how to set various window properties with vi editor.
3. Demonstrate how to auto indent lines and insert line numbers for the file.
4. Discuss the basic elements of communication process.
5. Demonstrate how UNIX administrators communicate with other users using mail, wall, send, mesg, and ftp.

Answer Keys

Self-assessment Questions	
Question No.	Answer
1	b
2	c
3	b
4	d
5	a
6	b
7	c
8	a
9	a
10	b
11	c
12	d
13	d

Activity

Activity Type: Offline **Duration: 30 Minutes**

Description:

1) Explain how one can send or receive messages using the mail command.

2) Do an online study and identify the advantages of email systems over the use of communication utilities that were developed earlier to them? If yes, mention and explain very briefly.

Bibliography

e-References

- This website was referred on 3rd May 2016, while developing content for vi editor https://www.google.co.in/search?q=Debian+and+RPM+packages&oq=Debian+and+RPM+packages&aqs=chrome..69i57.564j0j8&sourceid=chrome&es_sm=93&ie=UTF-8#
- This website was referred on 3rd May 2016, while developing content for vi editor http://vimdoc.sourceforge.net/htmldoc/options.html
- This website was referred on 3rd May 2016, while developing content for vi editor http://www.emerson.emory.edu/services/editors/vi/vi.html
- This website was referred on 3rd May 2016, while developing content for vi editor http://linux.die.net/man/1/write
- This website was referred on 3rd May 2016, while developing content for vi editor https://smccd.mrooms.net/pluginfile.php/677522/mod_resource/content/1/UNIXLinux_Resource_Files/7-part2-admiinistration.4U.pdf
- This website was referred on 3rd May 2016, while developing content for vi editor http://www.authorstream.com/Presentation/dpm2991-1759697-session-communication-tools-v1/

External Resources

- Maurice J. Bach, The Design of Unix Operating System, (2010) Pearson Education
- S. Prata, Advance UNIX, a Programmer's Guide, (2011), BPB Publications, and New Delhi,
- B.W. Kernighan & R. Pike, The UNIX Programming Environment, (2009) Prentice Hall of India.
- Jack Dent Tony Gaddis, Guide to UNIX Using LINUX, (2010) Vikas/ Thomson Pub. House Pvt. Ltd.

Video Links

Topic	Link
The Linux File System	https://www.youtube.com/watch?v=2qQTXp4rBEE
Directory structure of the UNIX file system	https://www.youtube.com/watch?v=PEmi550E7zw

Notes: