

Lab 4: Using the vi Editor

Step 1: Log in to Unix

Log-in with the user name `jiit` and password `jiitjiit`

Step 2: Access the Command Line

Select **Terminal** from the task bar to open a terminal window.

Step 3: Review the Three vi Modes

There are three modes of operation in vi Editor. Understanding the function of these three modes is the key to working with vi Editor. All commands available with vi Editor can be classified in one of the three modes. The table below lists the modes and a brief description of each. Review the three modes and answer the following questions.

Mode	Function / Characteristics
Command mode	Initial default mode for creating and editing files, cursor positioning and modification of existing text. All commands are initiated from this mode.
Insert mode	Used for entry of new text. Entering an insert command such <code>i</code> (insert), <code>a</code> (append) and <code>o</code> (open new line) will take the user from command mode to entry mode. Entry commands are stand-alone and are entered without pressing the Enter key.
Last-line mode	Used for saving work and quitting vi Editor. Type a colon (<code>:</code>) to get to this mode. Pressing the Enter key or Esc key returns to command mode.

1. Which vi mode is primarily used to enter new text?

2. Which vi mode is used to save the work and quit vi Editor?

3. When starting start the vi Editor, which mode is the default?

Step 4: Review Keystrokes to Move Between vi Modes

The table below shows how to switch modes and get from one mode to another. Review the keystrokes required to move between modes and answer the following questions.

From Mode	To Mode	Commands / Keystrokes
Command	Entry	i (input), o (open new line), a (append to existing line)
Entry	Command	Press Esc (Escape)
Command	Last-line	Colon (:)
Last-line	Command	Press Esc or Enter
Entry	Last-line	Press Esc to return to Command mode, then enter a colon
Last-line	Entry	Press Enter or Esc to return to Command mode, then enter an insert command

1. Which single-character alphabetic commands will put vi in Entry mode?

2. Which key will return vi to Command mode from either Last-line or Entry mode?

3. Which command will put vi into Last-line mode from Command mode?

Step 5: Create a New File with vi Editor

The vi Editor is started from the command line. Whenever invoking vi Editor, the user is opening a file. The name of the file to be created or edited can be specified when the user starts vi Editor, a new file can be opened to be named later. It is common to start vi Editor and specify a file name. If the file exists, it is opened for editing. If the file does not exist, it is created.

Command Format: **vi** [option(s)] [filename]

1. If logged in as the root user, switch users to another account (su -). If another account is not known, ask the instructor for an available account. Verify that the student is in the home directory. What command was used?

2. Open a new file called **myvifile** using the command: **vi myvifile**. What does the vi document screen look like?

3. Press the lower case letter **i** to begin inserting text at the first line. Is there any indication on the screen that the user is in Insert Entry mode?

4. Type the student's name. If a mistake is made, do not try to correct it at this time. Was the student able to enter text in Insert mode?

5. Press the **Esc** key to leave Insert Entry mode and return to Command mode. Is there any indication on the screen that the user is back in Command mode?

6. Type a **colon** to go from Command mode to Last-line mode. Is the user now at the bottom left corner of the screen at a colon (:) prompt? If not, press **Esc** again, and enter another colon.

7. Type a lower case **w** (write - to save the new file), followed by a lower case **q** (quit - to exit the vi Editor) and press **Enter**. What was the result of the **wq** commands? Is the user back to the command prompt now?

8. The new file should be saved in the practice directory on the hard disk. Display a long listing of this directory to verify that the file is there (**ls -l**). How many bytes are in the file?

Step 6: Review the Save and Quit Commands

In the previous steps, the student created a file and saved it with the **w** (write) command and exited vi Editor with the **q** (quit) command. There are several Save and Quit commands available with vi Editor. The following table lists some of the more common ones.

Note: Save and Quit commands (except for ZZ) are entered only when in Last-line mode.

Save and Quit Commands

1. Most save and quit commands are entered in which mode?

2. Which command will allow the user to exit vi Editor and not save any of the changes?

3. Which two commands listed in the table perform the same function?

4. Which command allows the user to save the current file being edited under another name, and remain in vi Editor to continue editing the current file?

Step 7: Open an Existing File

If the user starts vi Editor and specifies an existing file name, it is opened for editing. Here the student will add some new text in Entry mode and try a few cursor positioning commands.

1. Open **myvifile**, which was created earlier using the command: **vi myvifile**. What does the vi document screen look like?

2. Position the cursor at the end of the name and press the lower case letter **a** to Append to the line that was typed earlier. Type some text, and then press **Enter** for a hard return and then type some more text. Enter about three lines of text this way. What mode is the user currently in?

3. Press the **Esc** key to leave Insert mode and return to Command mode. In Command mode, can the user position the cursor (with the arrow keys) for additional editing?

4. The user can move the cursor with the arrow keys while in the various entry modes and will still remain in that mode. The table below contains some of the more common cursor positioning keys and commands. If using the alphabetic commands while in an entry mode, they will be entered as text. The user needs to press **Esc** to get to command mode to use them. Practice using these while editing this file. Which character moves the user back one word at a time?

Step 8: Cursor Positioning Commands

Command	Meaning
j or Down arrow	Move down one line
k or Up arrow	Move up one line
Space bar	Move right (forward) one character
w	Move forward one word (including punctuation)
Back Space	Move left one character(check for accuracy)
B	Move back one word (including punctuation)
\$	Move to end of line
0 (zero) or ^	Move to beginning of line
Return	Move down to beginning of next line

1. Press the up arrow to position the cursor until it is on line two. Use the **j** (jump down) command to move down and then use the **k** (kick up) command to move back to line two.

2. Type a lower case **O** to open a new line below the line the cursor is on and enter some new text. What mode is the user in now?

3. Press the **Esc** key again to leave Open line Entry mode and return to Command mode.

4. Type a colon (:) to go to Last-line mode then save and quit this file when finished. If this were a real file, and some mistakes had been made, and the user did not want to save the changes, what Last-line mode command would be used?

5. The new file should be saved in the practice directory on the hard disk. Display a long listing of this directory to verify that the file is there. How many bytes are in the file?

Step 9: Use Editing Commands

There are a many editing commands that can be used to modify existing text in a file. These include commands for deleting, and changing text. The majority of these commands are entered while in command mode.

1. Open a new document with vi Editor. What is the file named?

2. Insert some text. Add five or more lines of text and press **Enter** at the end of each line. Make some mistakes while typing.

3. Delete some text. While in command mode, position the cursor to the desired location and use the options shown in table below to delete some of the mistakes.

Note: These commands are ALL lower case and are entered without the Enter key.

Basic Text Deletion Commands

Command	Meaning
x	Delete character at the cursor
dw	Delete word (or part of word to right of cursor)
3dw	Delete three words
dd	Delete line containing the cursor
3dd	Delete three lines

1. Undo and change some text. To change text and undo changes, use the commands shown in the table below. Many of these commands change the user to Insert mode until Escape is pressed.

Note: These commands are ALL lower case.

Undo and Change Commands

Command	Meaning
cw	Change word (or part of word) at the cursor location to the end of the word
3cw	Change three words
r	Replace character at cursor with one other character
u	Undo previous command

1. Copy and paste text. To copy and paste text, use the following options:

Copy and Paste Commands

Command	Meaning
yy	Yank a copy of line and place in the clipboard
p	Put (paste) yanked or deleted line below current line
P (upper case)	Put (paste) yanked or deleted line above current line

1. Save the file and quit vi Editor.
What command was used?

Step 10: Customize the Session

The vi Editor includes options for customizing the edit sessions. These are summarized in the table below. The **set** command is used from last-line mode to control these options. The **set nu** option shows line numbers and is used frequently when writing programs and script files. The line numbers are not part of the file.

1. Open the document you just created with vi Editor. Use the commands in the table to customize the vi session.

Session Customizing Commands

Command	Meaning
:set nu	Show line numbers
:set nonu	Hide line numbers
:set showmode	Display current mode of operation
:set noshowmode	Turn off mode display
:set	Display all vi variables set
:set all	Display all possible vi variables and their current settings

Step 11: Use Search Commands

The commands in the table below allow the user to perform advanced editing such as finding lines or conducting searches for text. Note that the forward slash (/) and the question mark (?) search options are also last-line commands, but they do not require a colon first. The **next (n)** and **next previous (N)** commands can be used to find the next occurrence after the **/string** or **?string** commands found what the user was looking for.

Basic Search Commands

Command	Meaning
G (upper case)	Go to last line of file
:21	Go to line 21
/string	Search forward for string
?string	Search backward for string
n	Find next occurrence of string

Use the commands in the table to practice searching for a particular line or string of text. Quit vi Editor without saving any changes. What command was used?
