

**Fundamentals of UNIX**  
**Lab 9.1.8– Using the vi Editor**  
***(Estimated time: 45 min.)***

**Objectives:**

- Become familiar with the UNIX vi text editor
- Review the three vi Modes
- Review keystrokes to move between vi modes
- Create a new file with vi
- Invoke vi with showmode
- Review the save and quit commands
- Open an existing file with vedit
- Use editing commands
- Customize your session
- Use search commands

**Background:**

In this lab, you will use a UNIX text editing tool: the vi (pronounced "vee eye") editor. This text editor is primarily used for creating and modifying files that customize your work environment and for writing script files to automate tasks. System administrators use text editors to create and modify system files used for networking, security, application sharing, etc. The vi editor became a part of the UNIX operating system shortly after UNIX's inception and is universally available with UNIX systems. The vi editor is a very flexible and powerful editor with many options. These will be reviewed here with examples of their use.

For users learning to become system administrators, it is important to know how to use vi. It is sometimes the only full screen editor available to edit crucial system files. Examples of these include scripts and environment control files. Skill in using vi is also needed if the windowing system is not available. The vi editor is a useful tool when working remotely on other UNIX workstations or servers. Administrators routinely remote login or telnet to another UNIX computer to perform maintenance and troubleshooting tasks using vi. The availability and operation of vi is consistent across all UNIX platforms.

**Tools / Preparation:**

- a) Before starting this lab, review Chapter 9, Section 1 – The vi Editor
- b) You will need the following:
  - 1. A login user ID (e.g. user2) and password assigned by your instructor.
  - 2. A computer running the UNIX operating system
  - 3. Networked computers in classroom with class file system installed

**Notes:**

---

---

---

---

---

---

---

---

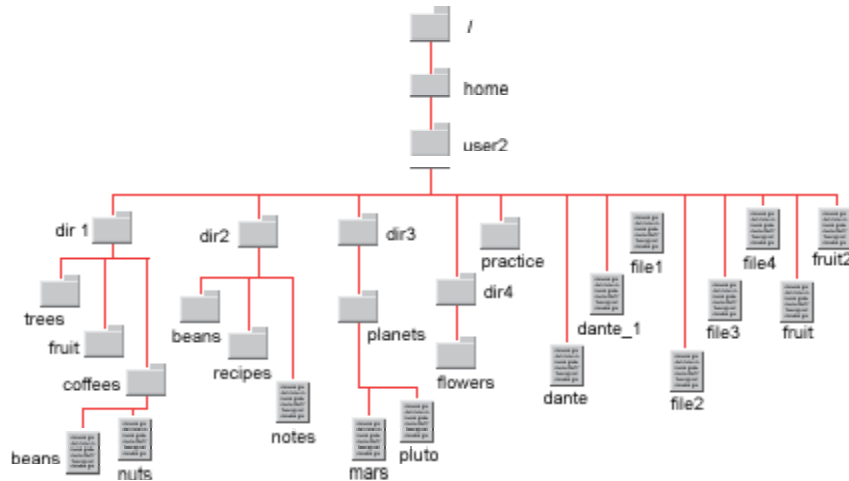
---

---

**Fundamentals of UNIX**  
**Lab 9.1.8– Using the vi Editor**  
**Worksheet**

Use the diagram of the sample Class File System directory tree to assist with this lab.

**Class File Tree Structure**



**Step 1. Log in to CDE**

Login with the user name and password assigned to you by your instructor in the CDE entry box.

**Step 2. Access the Command Line**

Right click on the **workspace** backdrop and click on **Tools**. Select **Terminal** from the menu to open a terminal window.

**Step 3. Review the Three vi Modes**

There are three modes of operation in vi. Understanding the function of these three modes is the key to working with vi. All commands available with vi can be classified in one of the three modes. The table below lists the modes and a brief description of each. Review the 3 modes and answer the following questions.

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

- Which vi mode is primarily used to enter new text? \_\_\_\_\_
- Which vi mode is used to save your work and quit vi? \_\_\_\_\_
- When you start the vi editor, which mode is the default? \_\_\_\_\_

**Fundamentals of UNIX**  
**Lab 9.1.8– Using the vi Editor**  
**Worksheet – Cont.**

**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

- Which single-character alphabetic commands will put vi in **Entry** mode? \_\_\_\_\_
- Which key will return vi to **Command** mode from either **Last-line** or **Entry** mode? \_\_\_\_\_
- Which command will put vi into **Last-line** mode from **Command** mode? \_\_\_\_\_

**Step 5. Create a new File with vi**

The vi editor is started from the command line. Whenever you invoke vi, you are opening a file. You can specify the name of the file you want to create or edit when you start vi or you can open a new file to be named later. It is common to start vi 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]**

- Change from your **home** directory to the **practice** directory. Verify that you are in the practice directory. What command did you use? \_\_\_\_\_
- Open a new file called **myvifile** using the command: **vi myvifile**. What does the vi document screen look like? \_\_\_\_\_
- Press the lower case letter **I** to begin **Inserting** text at the first line. Is there any indication on the screen that you are in Insert Entry mode? \_\_\_\_\_
- Type your name. If you make a mistake do not try to correct it at this time. Were you able to enter text in Insert mode? \_\_\_\_\_
- Press the **Esc** key to leave **Insert Entry** mode and return to **Command** mode. Is there any indication on the screen that you are back in Command mode? \_\_\_\_\_
- Type a **colon** (shift + :) to go from **Command** mode to **Last-line** mode. Are you now at the bottom left corner of the screen at a colon (:) prompt? \_\_\_\_\_ If not, press Esc again, and enter another colon.
- Type a lower case **W** (write - to save your new file), followed by a lower case **Q** (quit - to exit the vi editor). What was the result of the **wq** commands? \_\_\_\_\_  
What is your prompt now? \_\_\_\_\_

**Fundamentals of UNIX**  
**Lab 9.1.8– Using the vi Editor**  
**Worksheet – Cont.**

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

**Step 6. Invoke vi with Showmode**

In the previous step you noted that there was no indication as to what mode you were in. The **showmode** option displays the current mode in the lower right corner of the screen when you are in any of the Entry modes (Insert, Append, or Open). You can start vi and enter the Last-line command **set showmode** to turn showmode on or you can use **vedit** instead of **vi**. Vedit starts the vi editor with **showmode** turned on. If you are in **command** mode, nothing is displayed. If you are in **last-line** mode your cursor will usually be in the lower left corner with a colon displayed. Use **vedit** when possible so you will know what mode you are in. The remaining exercises will use **vedit** instead of **vi**.

- a. Open another new file called **myvifile2** using the command: **vedit myvifile2**. What does the vi document screen look like? \_\_\_\_\_
- b. Press the lower case letter “i” to begin **Inserting** text at the first line. What is displayed in the lower right corner of the screen? ? \_\_\_\_\_
- c. Press the **Esc** key to leave **Insert Entry** mode and return to **Command** mode. Is there any indication on the screen that you are back in Command mode? \_\_\_\_\_
- d. Press the lower case letter “o” to Open a new line. What is displayed in the lower right corner of the screen? ? \_\_\_\_\_
- e. Press the **Esc** key to leave **Open Entry** mode and return to **Command** mode. Is there any indication on the screen that you are back in Command mode? \_\_\_\_\_
- f. Press the lower case letter “a” to Append to the current cursor position. What is displayed in the lower right corner of the screen? ? \_\_\_\_\_
- g. Press the **Esc** key to leave **Append Entry** mode and return to **Command** mode. Is there any indication on the screen that you are back in Command mode? \_\_\_\_\_
- h. Type a **colon** (shift + :) to go from **Command** mode to **Last-line** mode. Are you now at the bottom left corner of the screen at a colon (:) prompt? \_\_\_\_\_ If not press Esc again and enter another colon.
- i. Type a lower case “q” (quit - to exit the vi editor) and an exclamation mark (!) to quit immediately and ignore any entries you may have made. What is your prompt now?
- \_\_\_\_\_

**Fundamentals of UNIX**  
**Lab 9.1.8– Using the vi Editor**  
**Worksheet – Cont.**

**Step 7. Review the Save and Quit Commands**

In the previous steps, you created a file and saved it with the **w** (write) command and exited vi with the **q** (quit) command. You also exited vi without saving any changes using the **q!** command. There are several Save and Quit commands available with vi. 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**

Command	Meaning
<b>:w</b>	Write buffer (Save changes and continue working in vi)
<b>:w new_filename</b>	Write buffer to new_filename (continue working in vi)
<b>:wq</b>	Write buffer (save changes) and quit vi
<b>ZZ (upper case)</b>	Save changes and quit vi. Alternative to :wq
<b>:q!</b>	Quit without saving changes
<b>:wq!</b>	Write buffer (save changes) and quit vi (The ! will override read only permissions if you are the owner of the file.)

- Most save and quit commands are entered in which mode? \_\_\_\_\_
- Which command will allow you to exit vi and not save any of you changes? \_\_\_\_\_
- Which two commands listed in the table perform the same function? \_\_\_\_\_
- Which command allows you to save the current file you are editing under another name and remain in vi to continue editing the current file? \_\_\_\_\_

**Step 8. Open an Existing File with vedit**

If you start vi or vedit and specify an excising file name, it is opened for editing. Here you will add some new text in **Entry** mode and try a few **cursor positioning** commands.

- Open **myvifile** which you created earlier using the command: **vedit myvifile**. What does the vi document screen look like? \_\_\_\_\_
- Position the cursor at the end of your name and press the lower case letter **A** to Append to the line you typed earlier. Type some text, and the press **Enter** for a hard return and then type some more text. Enter about three lines of text this way. What mode are you currently in?  
\_\_\_\_\_
- Press the **Esc** key to leave **Append** mode and return to **Command** mode. In Command mode, can you position your cursor for additional editing? \_\_\_\_\_
- You can move your cursor with the arrow keys while in the various entry modes and you will remain in that mode. The table below contains some of the more common **cursor positioning** keys and commands. If you use the alphabetic commands while in an entry mode they will be entered as text. You need to press Esc to get to command mode to use them. Practice using these while you are editing this file. Which character moves you back one word at a time?  
\_\_\_\_\_

**Fundamentals of UNIX**  
**Lab 9.1.8– Using the vi Editor**  
**Worksheet – Cont.**

**Cursor Positioning Commands**

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

- e. 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.
- f. Type a lower case **O** to **Open** a new line below the line your cursor is on and enter some new text. What mode are you in now? \_\_\_\_\_
- g. Press the **Esc** key again to leave **Open line Entry** mode and return to **Command** mode.
- h. Type a colon (:) to go to **Last-line** mode to **save** this file and **quit** when you are finished. If this were a real file and you had made some mistakes and did not want to save the changes you made, what Last-line mode command would you use? \_\_\_\_\_
- i. The new file should be saved in your practice directory on the hard disk. Display a long listing of this directory to verify that you file is there. How many bytes if 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.

- a. Open a new document with **vedit**. What did you name the file? \_\_\_\_\_
- b. Insert some text – Add 5 or more lines of text and press enter at the end of each line. Make some mistakes as you type.
- c. **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 your mistakes. **Note:** These commands are **ALL lower case** and are entered **without the Enter key**.

**Basic Text Deletion Commands**

Command	Meaning
<b>X</b>	Delete character at the cursor
<b>dw</b>	Delete word (or part of word to right of cursor)
<b>3dw</b>	Delete three words
<b>dd</b>	Delete line containing the cursor
<b>3dd</b>	Delete three lines

**Fundamentals of UNIX**  
**Lab 9.1.8– Using the vi Editor**  
**Worksheet – Cont.**

**d. Undo and Change some text** - To change text and undo changes, use the commands shown in the table below. Note: Many of these commands change you to Insert mode until you press Escape. **Note:** These commands are **ALL lower case**.

Undo and Change Commands	
Command	Meaning
<b>cw</b>	Change word (or part of word) at the cursor location to the end of the word
<b>3cw</b>	Change three words
<b>R</b>	Replace character at cursor with one other character
<b>U</b>	Undo previous command

**e. - Copy and paste text** – To copy and paste text, use the following options

Copy and Paste some text	
Command	Meaning
<b>yy</b>	Yank a copy of line
<b>p</b>	Put yanked or deleted line below current line
<b>P (upper case)</b>	Put yanked or deleted line above current line

**f. – Save the file and quit vi.** What command did you use? \_\_\_\_\_

**Step 10. Customize Your Session**

The vi editor includes options for customizing your 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.

a. Open the document you just created with **vedit**. Use the commands in the table to customize your vi session.

Session Customizing Commands	
Command	Meaning
<b>:set nu</b>	Show line numbers
<b>:set nonu</b>	Hide line numbers
<b>:set showmode</b>	Display current mode of operation
<b>:set noshowmode</b>	Turn off mode display
<b>:set</b>	Display all vi variables set
<b>:set all</b>	Display all possible vi variables and their current settings

**Fundamentals of UNIX**  
**Lab 9.1.8– Using the vi Editor**  
**Worksheet – Cont.**

**Step 11. Use Search Commands**

The commands in the table below allow you 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 you were looking for.

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

- a. Use the commands in the table to practice searching for a particular line or string of text.
- b. Quit vi without saving any changes. What command did you use? \_\_\_\_\_

**Step 12 – Remove Files and Directories Created in this Lab**

**Step 13. Close the Terminal Window and Logout**

Double click on the dash button in the upper left corner of the screen, then click the **EXIT** icon on the front panel.