**COMP 2766: Introduction to Linux**

**vi editor Lab**

**1.** Log into your system as a user named after your first name and last initial (ex: justint).

**2.** Enter this command line: **vi comp2766**

What is printed at the bottom of the screen?



**3.** When you first start the **vi** editor, you see tildes (~) down the left side of your screen. At the bottom left, you see the name of the file -- in this case, **"practiceFile" [New file]**. In this lab, you will be asked to type/enter information and press specific keys. The **vi** editor has three different functional modes. They are:

(a) **Input (a.k.a. insert)** mode. When you are in input mode, you are typing information into the file.

(b) **Command** mode. This is where keys are used as commands to act on text. The vi editor is a full screen editor and can edit the entire document to make changes.

(c) **Last Line** mode. This is where you can enter commands on the last line of the screen.

You will learn how to use these three modes in this lab.

**4.** Before you enter information in the file **comp2766**, issue the following command by entering everything that is shown in bold, here (***entering means pressing the Enter key after typing***):

**:set showmode**

or you can abbreviate the command by entering:

**:set smd**

A problem with the vi editor is that you usually cannot tell if you are in **insert (a.k.a. input) mode** or in **command mode**. The command, **:set showmode**, puts a message at the bottom left hand corner of your screen when you are in **insert** mode.

Note: Some versions of UNIX/Linux separate the **Input or Insert** mode into i**nsert, append, replace, change,** and **open** modes.

**5.** In order to see what the message looks like, enter insert mode by issuing the **vi** command that is shown in bold, here (***issuing means type, but do NOT press Enter***):

**i**

On the screen, you will not see the **i** that you typed. Instead, you should see a message on the bottom left of your screen.

What is the message that you see on the bottom left of your screen? — Insert —

**6.** The insert mode allows you to enter text into the file you are creating/editing. To make corrections, you sometimes have to exit **insert** mode. To do that, press the <**Esc**> key. You will see places in this lab where you are to press **<Esc>**. When are you finished with **insert** mode, press <**Esc**> and you will no longer see the message **--INSERT --** at the bottom left of your screen. That is how you know that you are in **command mode** and can enter commands to make changes to your file. To see how this works, press **<Esc>** now.

What happens to the message on the bottom of the screen? It disappeared.

**7.** Issue the following command:

**i**

You should be back in insert mode. The message, **-- INSERT --**, appears at the bottom left of your screen, again.

**8.** The other symbol that you will see frequently is the **<cr>** symbol. It stands for carriage return. Whenever you see **<cr>**, press the **Enter** key. **DO *NOT* TYPE <cr>**

**9. Typing in information using the vi editor**

Enter the following lines in bold. If you make a mistake, use the backspace and cursor keys to navigate to and fix the errors.

**BCIT<cr>**

**3700 Willingdon Ave<cr>**

After you have entered the above bolded lines, press <**Esc**>. The -- **INSERT**-- message should now be gone from the bottom left of the screen.

**10. Saving the Document**

Before you do anything else, save the document.

Issue the command: **:w**

What was printed on the screen?



The **w** means **write**. It is the command to save a file. You have saved the file, **comp2766**. Any changes you make to a file are only saved to a work buffer, whose contents disappear when the computer is powered off. If you are editing long files, save often to avoid having to re-enter everything if the computer goes down. Something to keep in mind when you are working with computers is:

**If something will go wrong, it will, and at the most inconvenient time!**

Let’s exit the vi editor and edit a different file.

Issue the command: **:q**

**11. Making Changes in the vi editor**

In your Linux virtual machine, load an Internet browser, login to Learning Hub, and download **practiceFile.txt** from Content, Session #6. The file should be in your user’s Downloads folder. Move it to your user’s home directory.

Open a terminal window and enter the following command line: **vi practiceFile.txt**

Upon loading the vi editor, you are in command mode and ready to make changes using commands. Note that, in the vi editor, a line is considered to be one line until <**cr**> is pressed. For example, if you typed an entire paragraph without pressing <**cr**>, you would have a paragraph that is considered one line long! This is significant when using keys/commands that move you up or down one line.

**12. Using command mode to make changes to your file**

You will be using various commands in this part of the lab. Linux is case sensitive, meaning that if you see a lowercase letter, you must use a lowercase letter and not an uppercase letter, as these have different meanings.

Let’s practice using command mode! **NOTE:** If you make a mistake, you can reverse the mistake by immediately issuing the undo command **u** and, if needed, you can issue it multiple consecutive times.

**13. Moving to the top or bottom of the document**

Sometimes, there are different ways to make the same changes. You can decide and choose the one(s) that you find easier to use. Move the cursor to the home line (top of document) of the work buffer by issuing the vi editor command:

**H**

The **H** takes you to the top of the document. After issuing **H**, which character in which word is highlighted by the cursor? The ‘T’ character of the first ‘The’ word on the first line.

Move the cursor to the last line of the work buffer by issuing the command:

**L**

In the case of the **H** and **L** commands, you did not press the enter key after you typed them. Which character in which word is highlighted now by the cursor? The ‘s’ character of the word ‘somewhere’ on the last line.

Note that **L** moves the cursor to the last line of the file, whereas, **l** (lower case L) moves the cursor right on the current line. Case matters in vi and Linux!

**14. Moving around in the document**

You can move one space or line at a time by using the **h**, **j**, **k**, and **l** keys. You should be at the bottom of the document now. Next, press **k** three times.

What happens when you do this? It goes three lines up.

Press the **l** (lower case L) key thrice (thrice means three times).

What happens when you do this? It goes three characters on the right.

Press the **j** key twice.

What happens when you do this? It goes three lines down.

Next, press the **h** key thrice.

What happens when you do this? It goes three characters on the left.

You can also move one word forward or backward. To do this, press **w** to move forward one word and **b** backward one word. Now, move to the top of the document.

How did you do this? Type H

Next press **w** four times.

What happened when you did this? It goes four words on the right.

Press **b** four times.

What happened when you did this? It goes four words on the left.

You have learned how to move to the top and bottom of the file. You also have seen how to move around in the file one character and one line at a time and how to move forward/backward one word.

NOTE: Depending on your computer setup and the terminal emulation, you may be able to use the arrow keys in place of the **h**, **j**, **k**, and **l** keys.

**15. Searching for words or sequences of letters**

Move to the top of the document by typing:

**H**

Now, you can search for certain words or sequences of characters. To do a forward search, meaning that the computer will search forward from the cursor to the bottom of the file, use **/** (forward slash) followed by the word or sequence of characters you are searching for. Find the word *fill* by using the forward search technique. Enter the following command:

**/fill**<**cr**>

Notice that you must press the Enter key after you type in this command.

Which character in which word is highlighted by the cursor? The word ‘fill’ on the first line.

Search forward, again, for the word *fill* by entering the command:

**/fi**<**cr**>

NOTE: For a forward search, the first line of the file follows the last line of the file. Also, since there is only one occurrence of the characters fi in the text, fi can be used as the search pattern instead of fill.

Now, move the cursor to the last line of the work buffer by typing:

**L**

Then, enter the backward search command:

**?fi**<**cr**>

Which character is the cursor on? It comes back on the ‘fi’ of the ‘fill’ word on the first line.

**NOTE:** For a backward search, the last line of the file follows the first line of the file.

Move the cursor to the last line of the work buffer.

Now, search backward through the text for the characters **and** (NOTE: there are several words with **and** in them). To repeat the find command, type in:

**n**

The **n** command searches for the next occurrence, so you can find all occurrences of the characters **and**. Go to the top of the document.

List in sequence all the words in the text that contained the characters **and**

‘and’ on lines : 2 (in word command), 7 13 and 14 (in word command).

**16. Making changes in words or adding new words**

Move to the word *fill* again. We are going to change *fill* to *full*. Use the **k** or **l** (lower case L) key to position the cursor over the *i*in the word *fill***.** With the cursor on the *i* in *fill*, issue the single character replace command:

**ru**

The character beneath the cursor is replaced by the character following **r**.

What happened to the word *fill* when you issued the **ru** command? It became ‘full’.

Next, change **full editor** to **full screen editor** by using either the insertcommand or the appendcommand. These are the differences between append and insert:

1. **Append** means "add to the end of." If you are at the end of a word and want to add on a new word, you can use append mode. If you were on the last *l* in the word *full* and you pressed the **a** key, you would move over one space to the right. Then you would start typing.

2. **Insert** mode is different. This is usually used when you want to add something before a word. If you were on the *f* in the word *full* and you pressed the **i** key, the cursor would stay on the letter *f*. You would, then, type a letter or word that would go in front of the letter *f*.

You need to move to the end of the word *full* or to the beginning of the word *editor* to make this change. Use the **h**, **j**, **k**, or **l** keys to do this. Once the cursor is in the correct position, use either **a** for append or **i** for insertto add the word *screen*. Type in:

**screen**

DON'T FORGET TO PRESS THE <**Esc**> KEY TO EXIT EITHER OF THESE MODES!

Which command(s) did you use? /fi —> e —> a —> ‘screen’

On which character of which word did you place the cursor before issuing the command? On the last l of ‘full’.

Try adding more words again by changing *k moves up* to *k moves the cursor up***.**

What command sequence did you issue to accomplish this change? 6j —> $ —> 2b —> ‘the cursor ‘

After you have made the above change, you need to make another one. Actually, the **k** key moves the cursor up only one line. You can change the word *two* to *one* by using the **cw** command. Position the cursor on the *t*in the word *two*. Then, type:

**cw** (do ***NOT*** press the enter key at this time)

What message do you see in the bottom left hand corner of your screen?



Also, do you see any other change to the word *two*? It disappeared.

Next, type in the word: **one**

Notice that you are still in CHANGE MODE. Press <**Esc**> to exit that mode and return to command mode.

Now that *two* has been changed to *one*, you need to change the word *lines* to *line*. Move to the *s* in the word *lines* and, then, use the **x** command. This deletes one character. Place your cursor on the *s* and, then, type in:

**x**

What happened when you typed **x** ? ‘s’ was deleted.

Next, position the cursor so that you can change *yhe* to *the*. Use the single character replace command.

What command sequence did you issue to accomplish this? k —> 7b —> rt

**17. Deleting words or lines**

Position the cursor on the *t* in the word *typing* and type:

**dw**

to delete the word *typing*. Now, undo the deletion by issuing the undo command:

**u**

If you placed the cursor on the *t* in *typing*, the entire word was deleted.

What would have happened if you had placed the cursor on the *y*in *typing*, instead?

Try it and report the results here: Only the characters from y (included) to the end of the word would be deleted.

Undo whatever you just did so that the word *typing* is still in the text. The word *The* is duplicated on the first line of the work buffer. Use the **dw** command to delete one of the duplicates.

Move the cursor to the number*(1)*. Then issue the command:

**dd**

What happened? The line was completely deleted.

Restore the text by issuing the undo command. Next move to the line that begins with the number **(2)**. Enter the command:

**d<cr>**

What happened this time? The line was completely deleted.

From these examples, what if any difference is there between **dd** and **d<cr>** ? No

Issue the undo command to restore the text.

Four lines of the text refer to deleting a word or the entire line. Position the cursor on the line that contains *Delete line 1*. Issue the command:

**3dd**

What happened as a result? Three lines (from where we was) was deleted.

Now issue the undo command:

**u**

What happened? They reappeared.

Now, delete the two lines containing the statements *Delete line*.

Next, position the cursor to delete one of the duplicated words (*word*and *delete***)** in the two remaining sentences.

What commands did you issue to accomplish this? I went on one of the duplicate word for each sentence and deleted the second occurence of the same word by dw them.

**18. Copying or moving lines to other locations in a document**

If you want to copy a line or lines to a new location in a document, you can do this by using the **y** for the yank command. The **p** for put is used to, then, place the line in the new location. Be sure that you are in command mode. Move to the line that starts with *(1)*. Issue the yank command:

**y** (a lowercase Y)

You will not see any change at this time. But the lines with the (1) and (2) will be kept in a buffer. Now, put the cursor at the beginning of the line that says *You can also move lines to a new location by using the "y" for yank.*Issue the put command:

**p** (a lowercase P)

What are the results? The lines (1) and (2) was added under the “You cal also...” sentence.

The **y** command will copy two lines. The **p** command should place two lines after the line listed above. Now, remove the two lines by entering:

**d**<**cr**>

Next, go to the line that starts with the *(1)*. Issue this command:

**yy** (lowercase YY)

Next, move the cursor to the line that starts with: *You can also move lines to a new location by using the "y" for yank.*Issue this command:

**P** (uppercase P)

What happened this time? It added the line (1) above the line “You can also...”

The **yy** will only yank one line just as the **dd** command only removes one line. The **P** command will put the line beforethe line containing the cursor.

You can also copy more than one line at a time, by using the **nyy** where the **n** is the number of lines you want to copy. Try this by moving to the beginning of the line:

*You can also move lines to a new location by using the "y" for yank.*

This time, type:

**3yy**

Next move to the line with the *(2)* in it. Place your cursor on that line. Issue the put command:

**p** (that’s a lowercase P)

What happened this time? The three copied lines (the three last one of the document) were pasted under the line (2).

After you copy lines to a new location, you can remove the other copies of the lines by using the **dd** or the **d** commands if you wish.

**19. Changing words and other error corrections**

Next, look over the text to correct any other errors present. Note that both the **h** and the **l** key say that they move to the left. Only one of them moves to the left--the "l" key moves to the right. You can change this by using the **cw** command as you used before. You can also use the **R** command to replace word. Position the cursor on the letter "l" in left. Then, type in:

**Rright**

What message do you see at the bottom of your screen?



What must you do to return to command mode? Esc command

Also the word, *currsor* is spelled wrong. Remove the extra *r* in *currsor*.

What command did you use to fix this problem? 5w —> 2l (lowercase L) —> x

Next, search for the pattern **ona**. There is supposed to be a space between *on* and *a*. Place your cursor on the letter *n* in *ona*. Next, type **a** for append.

What happens to the cursor and what message do you see on the bottom of the screen? It moved one character to the right (currently on ‘a’).

Press the space bar next. What key do you have to press to get out of the **Append Mode**? Esc command

**20. Adding new lines**

Next, move to the line that starts with *Delete word 1***.** Position the cursor on the *D* in *Delete*. Issue the open command:

**O** (that is, a capital O)

What happened when you typed **O**? A new line appeared above.

You have opened a new line above the line that you were on and you are placed in insert mode. You are now ready to enter these three lines:

**If you want to open a new line above, you can press the "O" key.<cr>  
If you want to open a new line below the line you are on, <cr>  
press the "o" (lower case "o") key. <esc>**

**21. Joining lines together**

Another feature that may be useful is to join two lines together. Go to the end of the line that says "If you want to open a new line below the line you are on,". Put the cursor on the "," on that line. Type in:

**J**

Note that this is the capital letter **J**.

What happens when you do this? The next line was moved right after the comma.

**22. Saving your file**

Next, save your file so it will be available to you the next time you need it. After correcting the errors, issue the command:

**:wq**

What was printed? Nothing because it makes us leave vim.

The **wq** stands for write and quit. You used **:w** earlier to save the file.

The original file was replaced with all the changes to it that you made.

You have used examples where you have been in the "INSERT MODE", "APPEND MODE", "CHANGE MODE", "REPLACE MODE", and also "OPEN MODE". After you have finished in any of these modes, you had to press the **escape** key to get out of them.

**23.** Submit to the appropriate dropbox this document with all your answers along with your edited **practiceFile.txt**