Experiment 4

Aim: Study of vi command

Theory:

The default editor that comes with the UNIX operating system is called vi (visual editor).

The UNIX vi editor is a full screen editor and has two modes of operation:

* Command mode commands which cause action to be taken on the file, and
* Insert mode in which entered text is inserted into the file.

In the command mode, every character typed is a command that does something to the text file being edited; a character typed in the command mode may even cause the vi editor to enter the insert mode. In the insert mode, every character typed is added to the text in the file; pressing the <Esc> (Escape) key turns off the Insert mode.

While there are several vi commands, just a handful of these is usually sufficient for beginning vi users. To assist such users, this Web page contains a sampling of basic vi commands. The most basic and useful commands are marked with an asterisk (\* or star) in the tables below. With practice, these commands should become automatic.

**NOTE:** Both UNIX and vi are case-sensitive. Be sure not to use a capital letter in place of a lowercase letter; the results will not be what you expect.

**To Get into and Out Of vi**

**To Start vi**

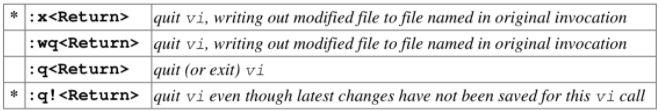
To use vi on a file, type in vi filename. If the file named filename exists, then the first page (or screen) of the file will be displayed; if the file does not exist, then an empty file and screen are created into which you may enter text.



**To Exit vi**

Usually, the new or modified file is saved when you leave vi. However, it is also possible to quit vi without saving the file.

Note: The cursor moves to bottom of screen whenever a colon (:) is typed. This type of command is completed by hitting the <Return> (or <Enter>) key.

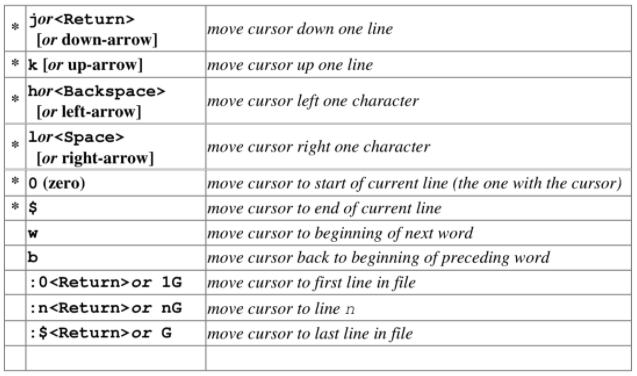


**Moving the Cursor**

Unlike many of the PC and MacIntosh editors, the mouse does not move the cursor within the vi-editor screen (or window). You must use the key commands listed below. On some UNIX platforms, the arrow keys may be used as well; however, since vi was designed with the Qwerty keyboard (containing no arrow keys) in mind, the arrow keys sometimes produce strange effects in vi and should be avoided.

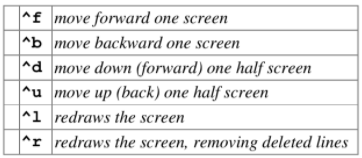
If you go back and forth between a PC environment and a UNIX environment, you may find that this dissimilarity in methods for cursor movement is the most frustrating difference between the two.

In the table below, the symbol ^ before a letter means that the <Ctrl> key should be held down while the letter key is pressed.



**Screen Manipulation**

The following commands allow the vi-editor screen (or window) to move up or down several lines and to be refreshed.



**Adding, Changing, and Deleting Text**

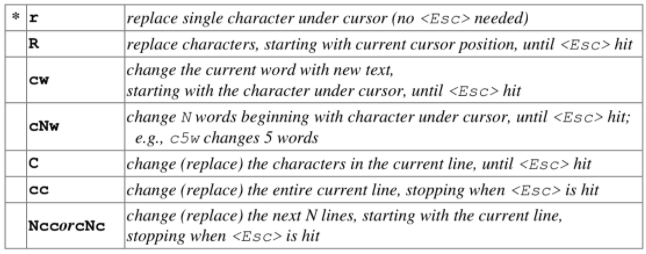
Unlike PC editors, you cannot replace or delete text by highlighting it with the mouse. Instead use the commands in the following tables.

Perhaps the most important command is the one that allows you to back up and undo your last action. Unfortunately, this command acts like a toggle, undoing and redoing your most recent action. You cannot go back more than one step.



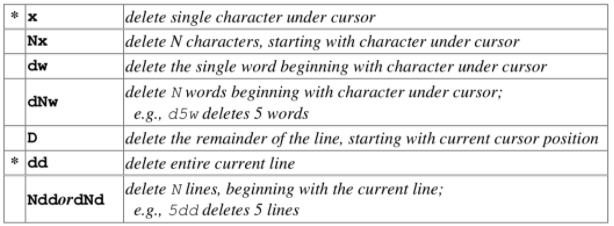
**Inserting or Adding Text**

The following commands allow you to insert and add text. Each of these commands puts the vi editor into insert mode; thus, the <Esc> key must be pressed to terminate the entry of text and to put the vi editor back into command mode.



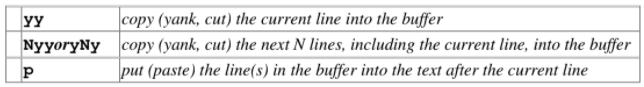
**Deleting Text**

The following commands allow you to delete text.



**Cutting and Pasting Text**

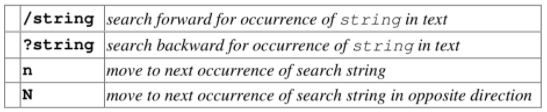
The following commands allow you to copy and paste text.



**Other Commands**

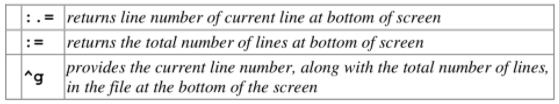
**Searching Text**

A common occurrence in text editing is to replace one word or phrase by another. To locate instances of sets of characters (or strings), use the following commands.



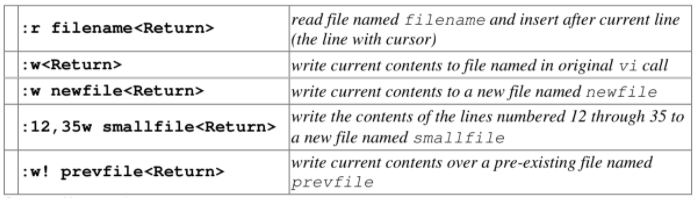
**Determining Line Numbers**

Being able to determine the line number of the current line or the total number of lines in the file being edited is sometimes useful.



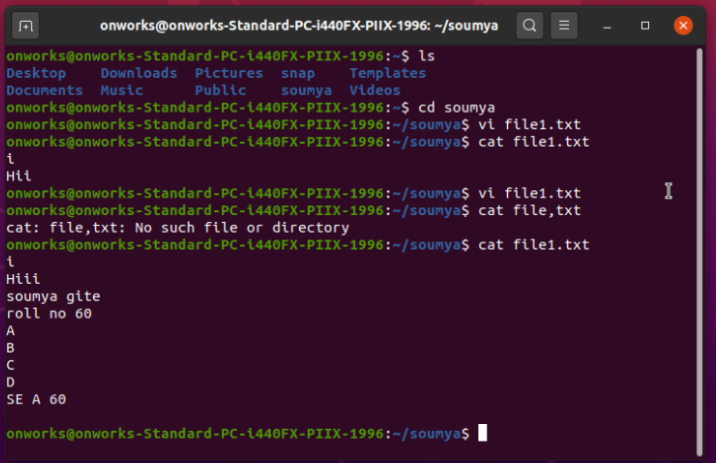
**Saving and Reading Files**

These commands permit you to input and output files other than the named file with which you are currently working.

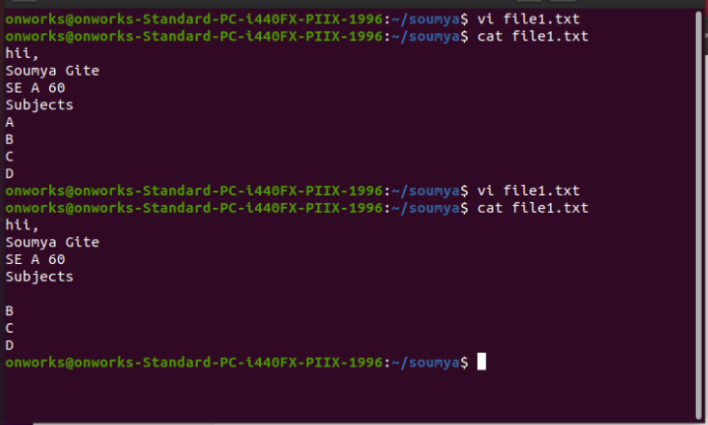


**Exceuting in terminal:**

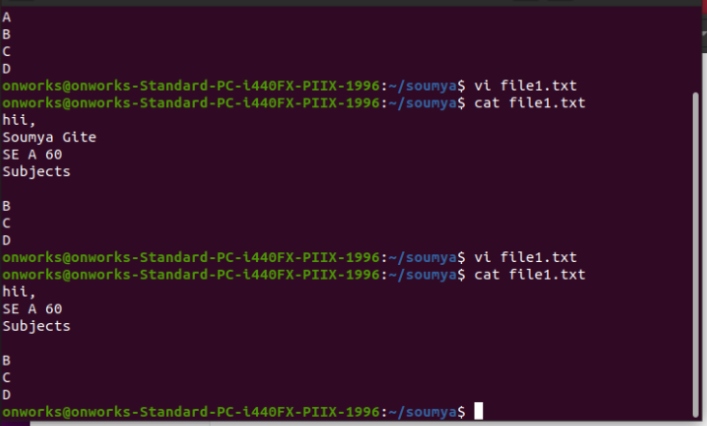
insert command (i)



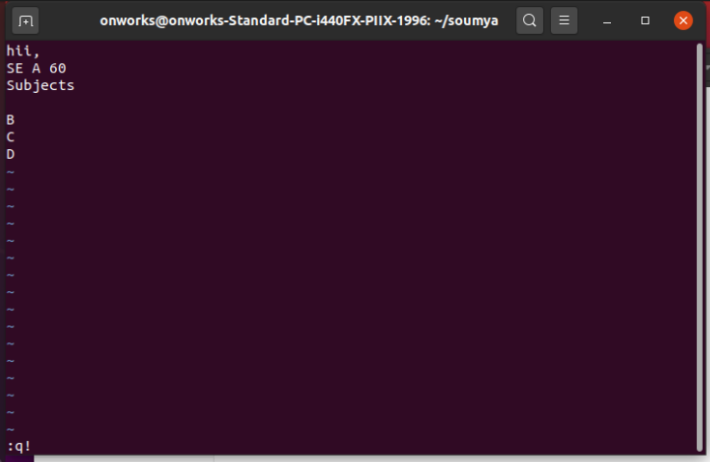
Deleting line using x command



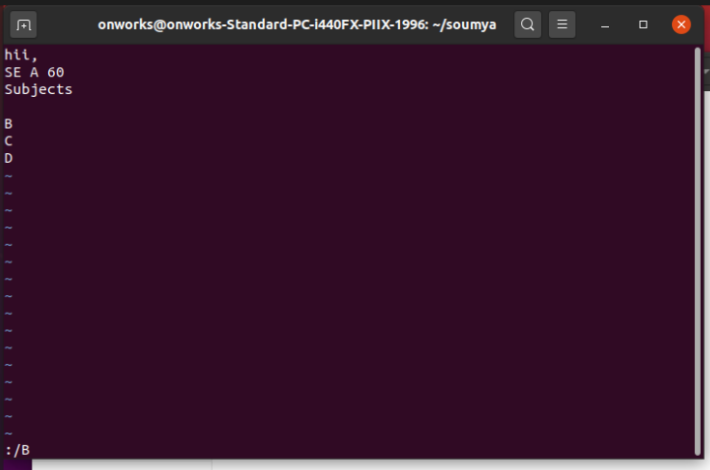
Deleting using dd command



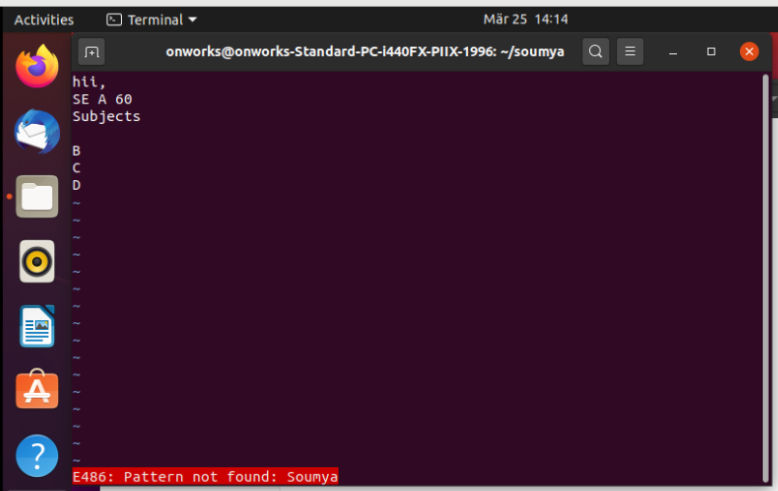
To exit vi using q! command



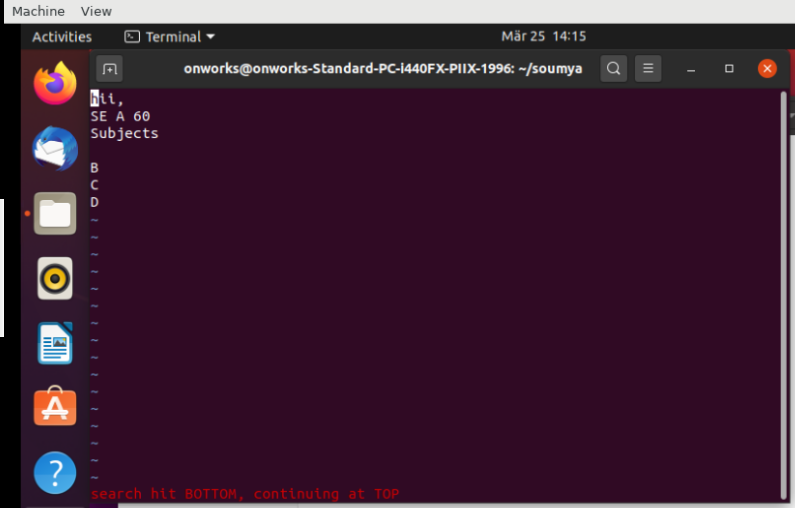
Searching



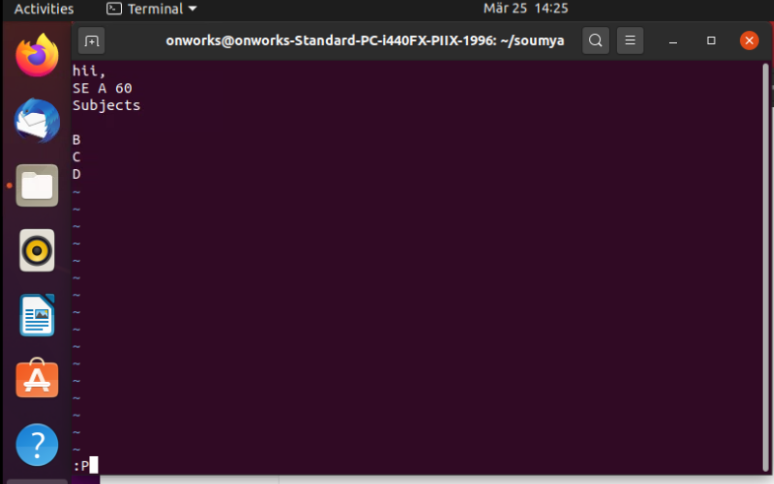
Search using(/string)

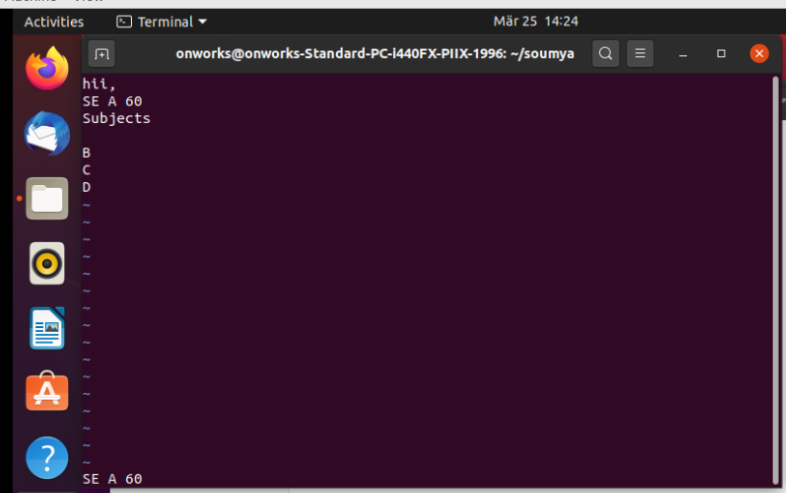


Search 1st line

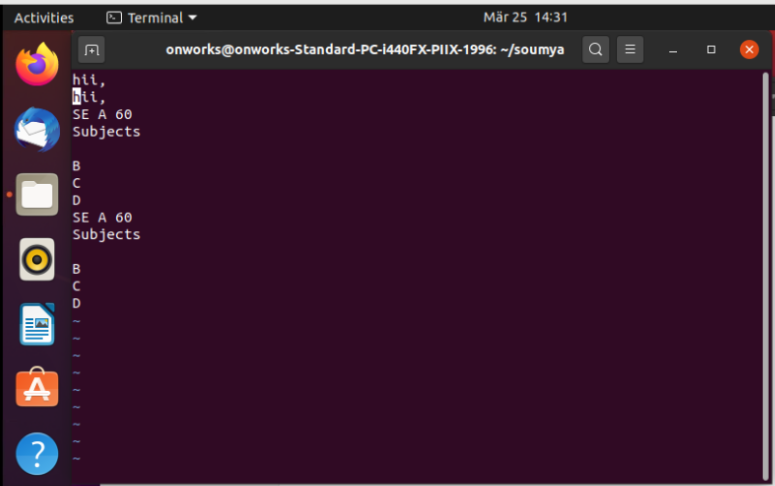


Paste using P command

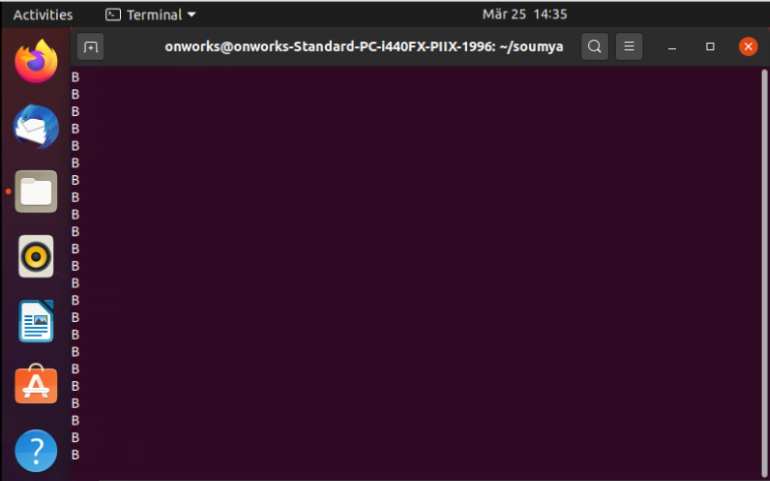




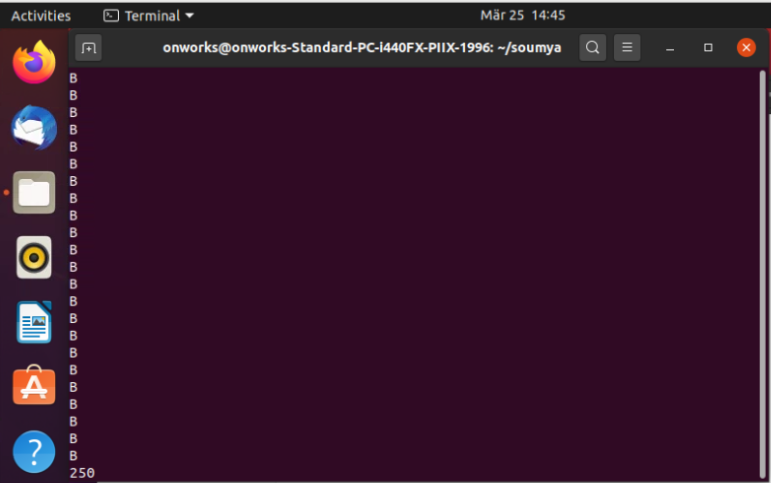
Copy



Screen manipulation



Determining using = command



Conclusion: Performing vi commands in terminal is successfully implemented and executed.