**Experiment 3**

**Date:** 7 - 01 - 2021

**Aim:** To explore advance Linux commands.

**Software Used:** Cgywin64 Terminal.

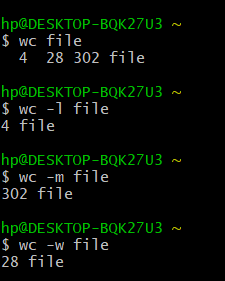
**Theory:**

1. **wc:** wc command is used for Used for printing newline, word and byte counts for files. It can return the number of lines in a file, the number of characters in a file and the number of words in a file. The output is number of lines, number of words, number of bytes, filename.

**Syntax:** wc <filename> or wc <options> <filename>.

**Note:** Options for wc command are:

* -l - To print the number of lines in a file.
* -m - To print the number of characters in a file.
* -w - To print the number of words in a file.



1. **cmp:** cmp command is used to compare the two files byte by byte. It helps you to find out whether the two files are identical or not. It reports the location of the first mismatch to the screen if difference is found and if no difference is found i.e the files compared are identical. It displays no message and simply returns the prompt if the files compared are identical.

**Syntax:** cmp <option> <filename> or cmp <filename>.

**Note:**

* -b - display the differing bytes in its output.

cmp -b file1 file2

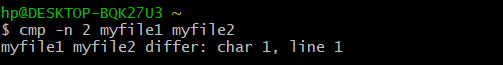
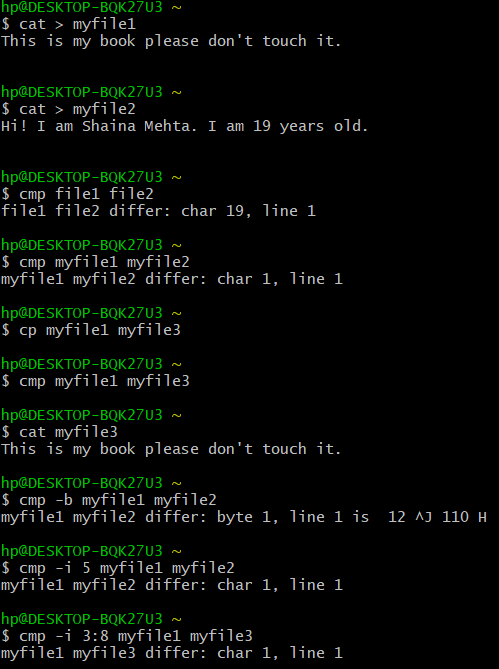
* -i [bytes-to-be-skipped] - Now, this option when used with cmp command helps to skip a particular number of initial bytes from both the files and then after skipping it compares the files.

cmp –i 5 file1 file2

* -i [bytes to be skipped from first file] : [bytes to be skipped from second file] - This option is very much similar to the above -i [bytes to be skipped] option but with the difference that now it allows us to input the number of bytes we want to skip from both the files separately.

cmp –i 4:4 file1 file2

* -n [number of bytes to be compared] option - This option allows you to limit the number of bytes you want to compare, like if there is only need to compare at most 25 or 50 bytes.



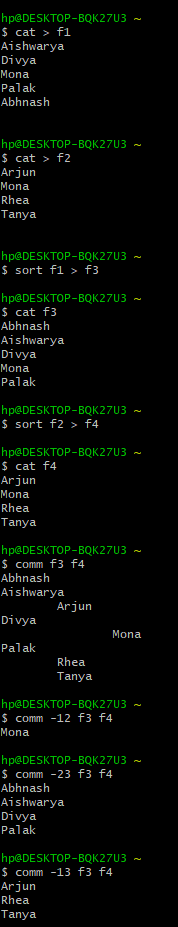
1. **comm:**

* It requires two sorted files and lists the differing entries in different columns.
* When you run comm, it displays a three – columnar output.
* The first column contains the lines unique to the first file, and the second column shows the lines unique to the second file. The third column displays lines to both files.

**Syntax:** comm <options> <sorted\_file\_1> <sorted\_file\_2> or comm <sorted\_file\_1> <sorted\_file\_2>.

**Note:**

* These commands require single column output from comm, and comm can produce using the options -1, -2 or -3.
* To drop a particular column simply use its column number as an option prefix.



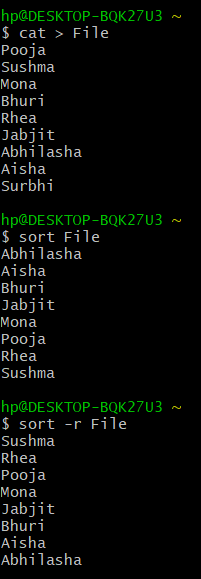
1. **sort:**

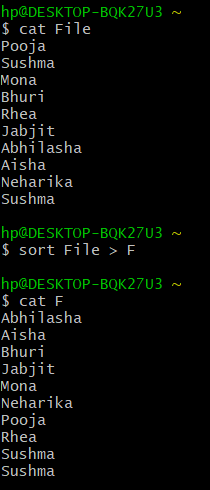
* sort lines alphabetically by default.
* Running sort filename writes the contents of the filename in alphabetical order to standard output.

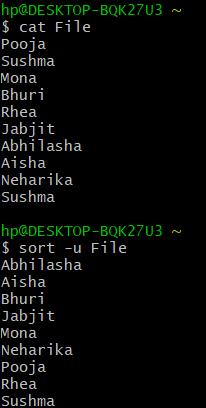
**Syntax:** sort <options> <filename> or sort <filename> or sort <file1> <file2>

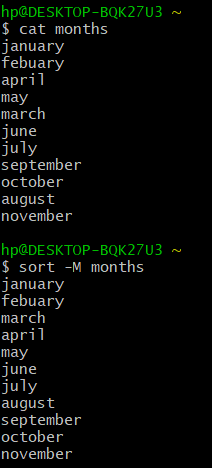
**Note:**

* + -r - sort in reverse order and write the result to standard output.
  + -n - This will sort from lowest number to highest number and write the result to standard output.
  + To sort and remove duplicates pass the -u option to sort. This will write a sorted list to standard output and remove duplicates.
  + To sort by month pass the -M option to sort.









1. **Creating Files in Linux:** It requires the use of an Editor.Various editors are used for this purpose which are:

* nano / pico
* vi
* emacs

**Vi Editor:**

* The VI editor is the most popular and classic text editor in the Linux family.
* Below, are some reasons which make it a widely used editor:
  + - * + available in almost all Linux Distributions
        + works the same across different platforms and Distributions
        + user-friendly

**Modes of Vi Editor:**

* + - * Command Mode
      * Insert Mode
      * Escape Mode

**Command Mode:**

* vi starts in Command Mode.
* vi interprets any characters we type as commands and does not display them in the window.
* This mode allows us to move through a file, and to delete, copy, or paste a piece of text.
* To enter into Command Mode from any other mode, it requires pressing the [Esc] key. If we press [Esc] when we are already in Command Mode, then vi will beep or flash the screen.

**Insert Mode:**

* Enables you to insert text into the file.
* Everything that’s typed in this mode is interpreted as input and finally, it is put in the file.
* The vi always starts in command mode. To enter text, you must be in insert mode. To come in insert mode, you simply type i. To get out of insert mode, press the Esc key, which will put you back into command mode.

**Escape Mode:**

* enables you to perform tasks such as saving files, executing commands.
* invoked by typing a colon [:], while vi is in Command Mode.
* The cursor will jump to the last line of the screen and vi will wait for a command.

**Vi Editor Commands:**

* i - Insert at cursor (goes into insert mode)
* a - Write after cursor (goes into insert mode)
* A - Write at the end of line (goes into insert mode)
* ESC - Terminate insert mode
* U - Undo all changes to the entire line
* - Open a new line (goes into insert mode)
* dd - Delete line
* 3dd - Delete 3 lines.
* D - Delete contents of line after the cursor
* dw - Delete word
* 4dw - Delete 4 words
* cw - Change word
* x - Delete character at the cursor
* r - Replace character
* R - Overwrite characters from cursor onward
* s - Substitute one character under cursor continue to insert
* S - Substitute entire line and begin to insert at the beginning of the line
* k - Move cursor up
* j - Move cursor down
* h - Move cursor left
* l - Move cursor right
* :w - Save the file but keep it open
* :q - Quit without saving
* :wq - Save the file and quit

**Note:**

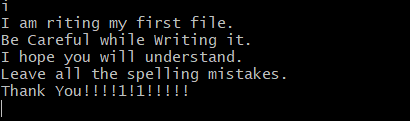
* Make sure you press the right command otherwise you will end up making undesirable changes to the file.
* You can also enter the insert mode by pressing a, A, o,

Opening of Vi Editor

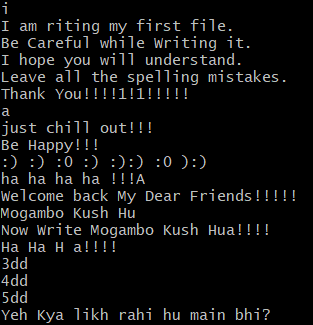




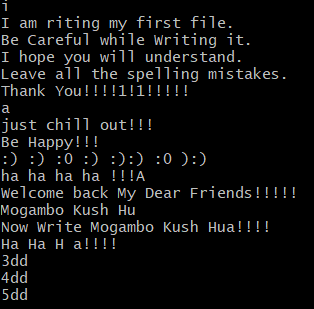
Insert at cursor (goes into insert mode)



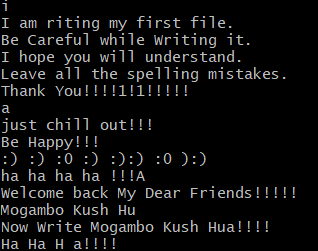
Write after cursor (goes into insert mode) and write at the end of line (goes into insert mode)



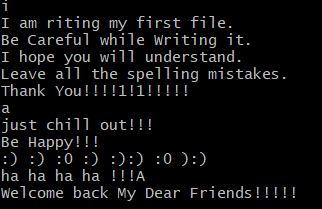
Delete line

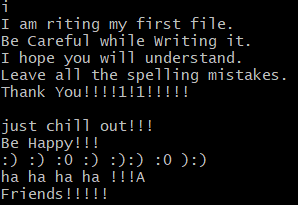


Delete 3 lines.

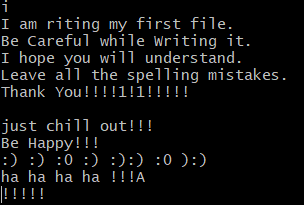


Delete 4 words

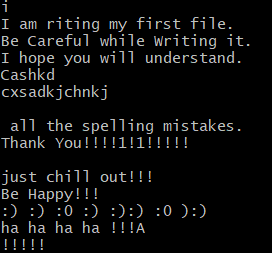


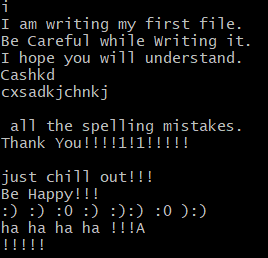


Delete word

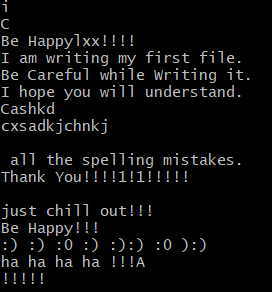


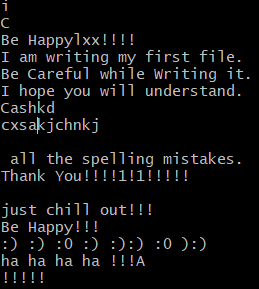
Change word



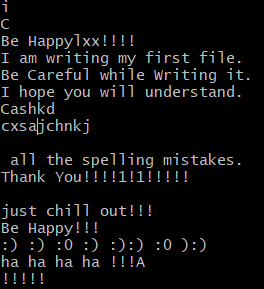


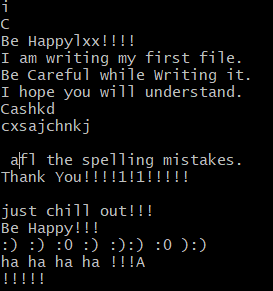
Delete character at the cursor



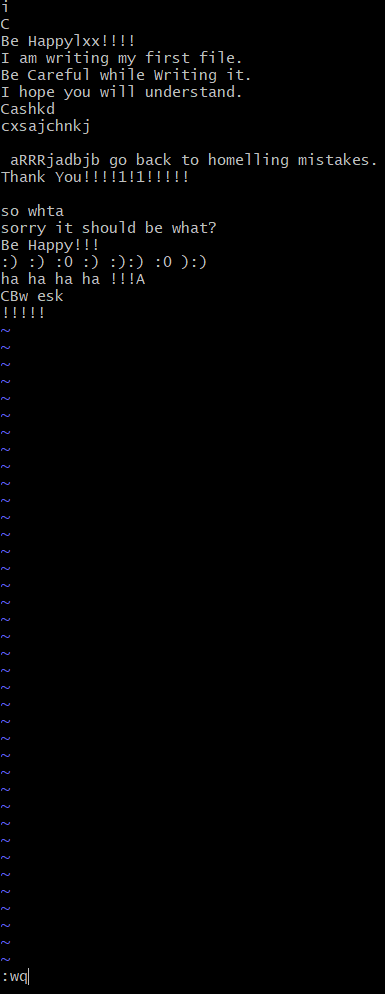


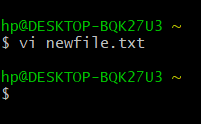
Replace character





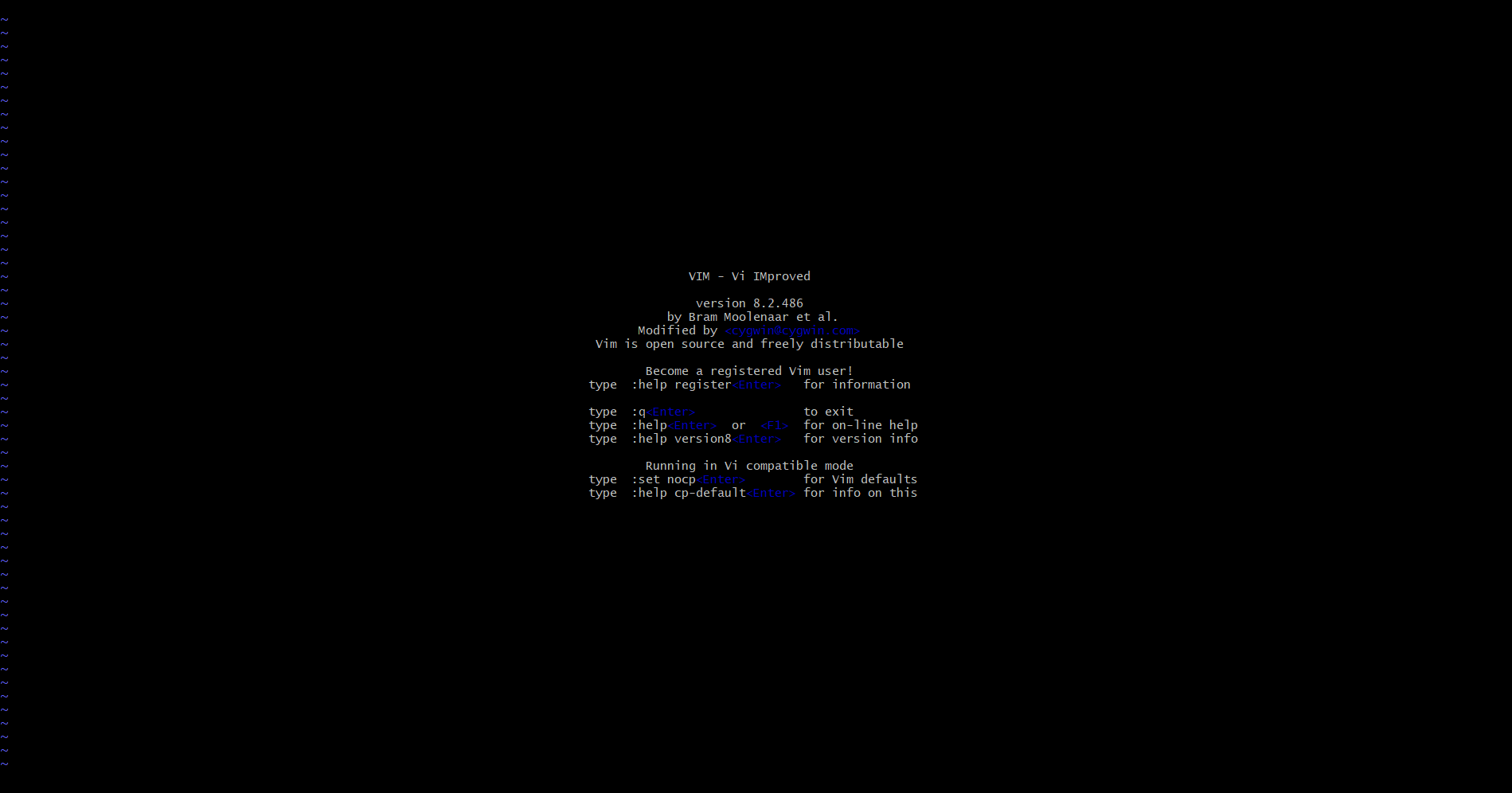
Save the file and quit





Type “vi” at the prompt





**Results:** Linux commands has been executed successfully.