# Experiment – 5

## Study of linux editors with their command specifications

Linux text editor is kind of a computer program that can edit text like notepad software. Text editors provide services in software development and operating system softwares which can be utilized to include programming languages, source code, documentation files etc.

Linux text editors can be used for editing text files, writing codes and updating user instruction files and many more.

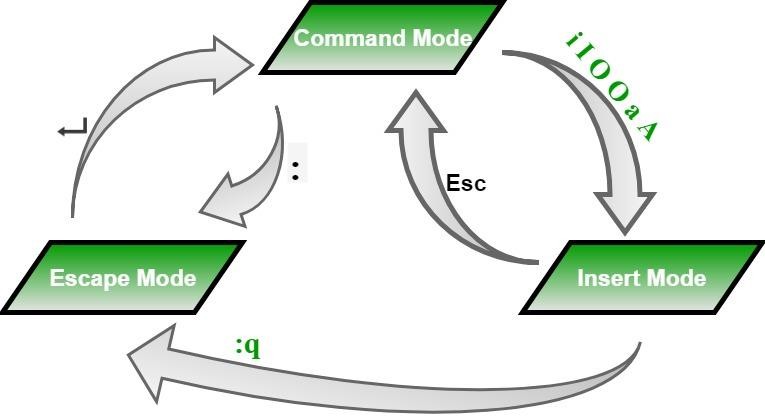
Linux Text editors come in two forms-

1. **Command Line Text editors** (such as Vi, nano, pico and more) and
2. **GUI based text editors** (such as GNOME, KDE, etc)

## Introduction of Vi editor :-

The default editor that comes with a UNIX os is Vi editor(Visual editor).

The unix editor is a full screen editor used to perform a variety of tasks. The unix architecture comes in three modes: **(i) Command**, **(ii) insert** and **(iii) Esc mode**



Modes of operation and their transition in VI editor

## Vim

### Overview:

Vim is a highly configurable text editor built to enable efficient text editing. It has a steeper learning curve compared to nano but offers powerful features.

### Basic Commands:

**Open a file:**

vi filename.txt

### Switch to insert mode:

Press i.

### Save changes and exit:

Press Esc, then type :wq and press Enter.

### Exit without saving:

Press Esc, then type :q! and press Enter.

### Delete a line:

Press Esc, then type dd.

### Undo changes:

Press u.

### Search text:

Press / followed by the search term, then press Enter.

## Features:

* + Multiple modes (normal, insert, visual).
  + Extensive plugins and configurations.
  + Powerful search and replace capabilities.

### Commands :-

In Command mode each character is typed as a command that does something to the text file. The basic vi editor commands are,

* 1. vi <file name>:- edit filename starting at line-1.



* 1. To insert text:-

i (before cursor)



* 1. To delete text:-
     1. dd



## Nano

### Overview:

Nano is a simple, user-friendly text editor that is easy to use for beginners.

### Basic Commands:

* + **Open a file:**

nano filename.txt

### Save changes:

Press Ctrl + O (then press Enter).

### Exit:

Press Ctrl + X.

### Cut text:

Press Ctrl + K.

### Paste text:

Press Ctrl + U.

### Search text:

Press Ctrl + W (then type the search term and press Enter).

## Features:

* + Simple interface.
  + On-screen command shortcuts.
  + Supports syntax highlighting.

## Emacs

### Overview:

Emacs is a versatile and extensible text editor, often used for programming and text manipulation. It provides a rich set of features and customization options.

### Basic Commands:

**Open a file:** emacs filename.txt



### Save changes:

Press Ctrl + X, then Ctrl + S.

### Exit:

Press Ctrl + X, then Ctrl + C.

### Cut text (kill):

Press Ctrl + K.

### Paste text (yank):

Press Ctrl + Y.

### Search text:

Press Ctrl + S (type the search term and press Enter).

**Summary of Commands**

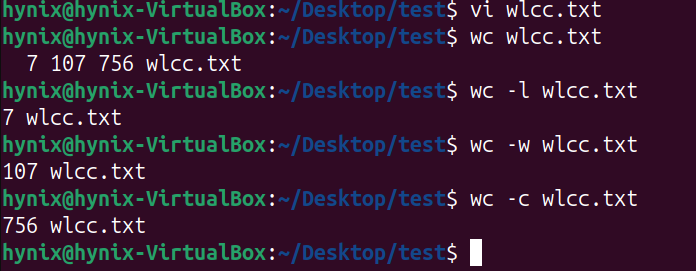
|  |  |  |  |
| --- | --- | --- | --- |
| Command | Nano | Vim | Emacs |
| Open a file | Nano filename | Vim filename | emacs filename |
| Save changes | ctrl+o | :w | Ctrl+x, ctrl+s |
| Exit | ctrl+x | :q | Ctrl+x, ctrl+c |
| Cut Text | ctrl+k | dd | ctrl+k |
| Paste Text | ctrl+u | p | ctrl+y |
| Search Text | ctrl+w | / | ctrl+s |
| Undo Changes | Not available | u | ctrl+\_ |

# Experiment-6

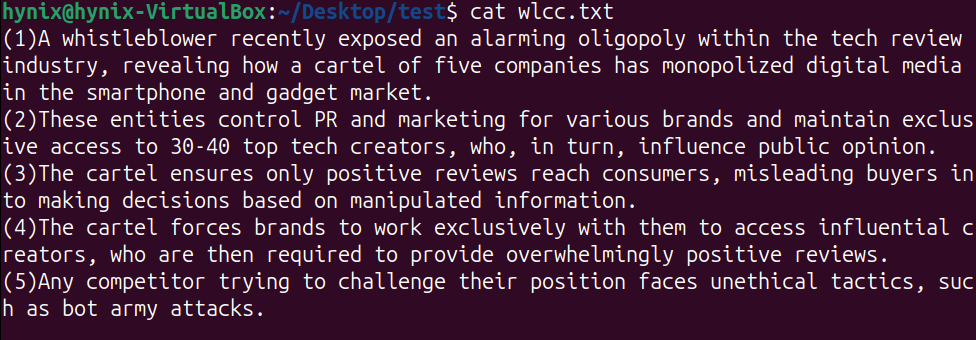
## Create a file called WLCC.txt with some lines and display how many lines, words and characters are present.

**Step-1**: vi command opens the text editor

**Step-2**: wc command is used to count the number of lines, words, and characters respectively.



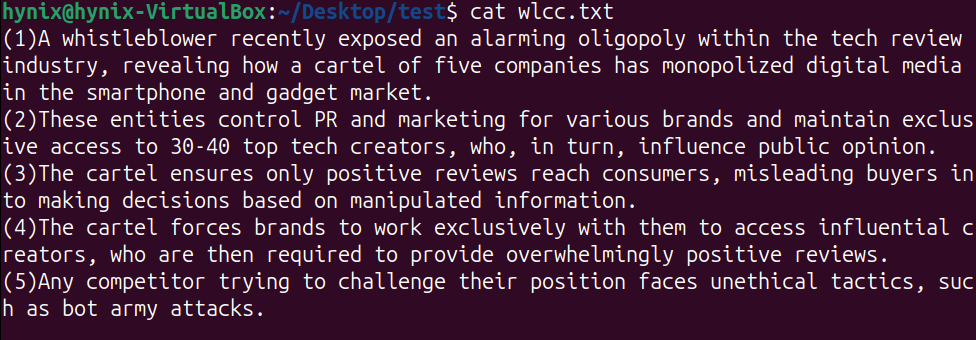
**Contents of wlcc.txt -**



# Experiment - 7

## Append 10 more lines in the file WLCC.txt and display

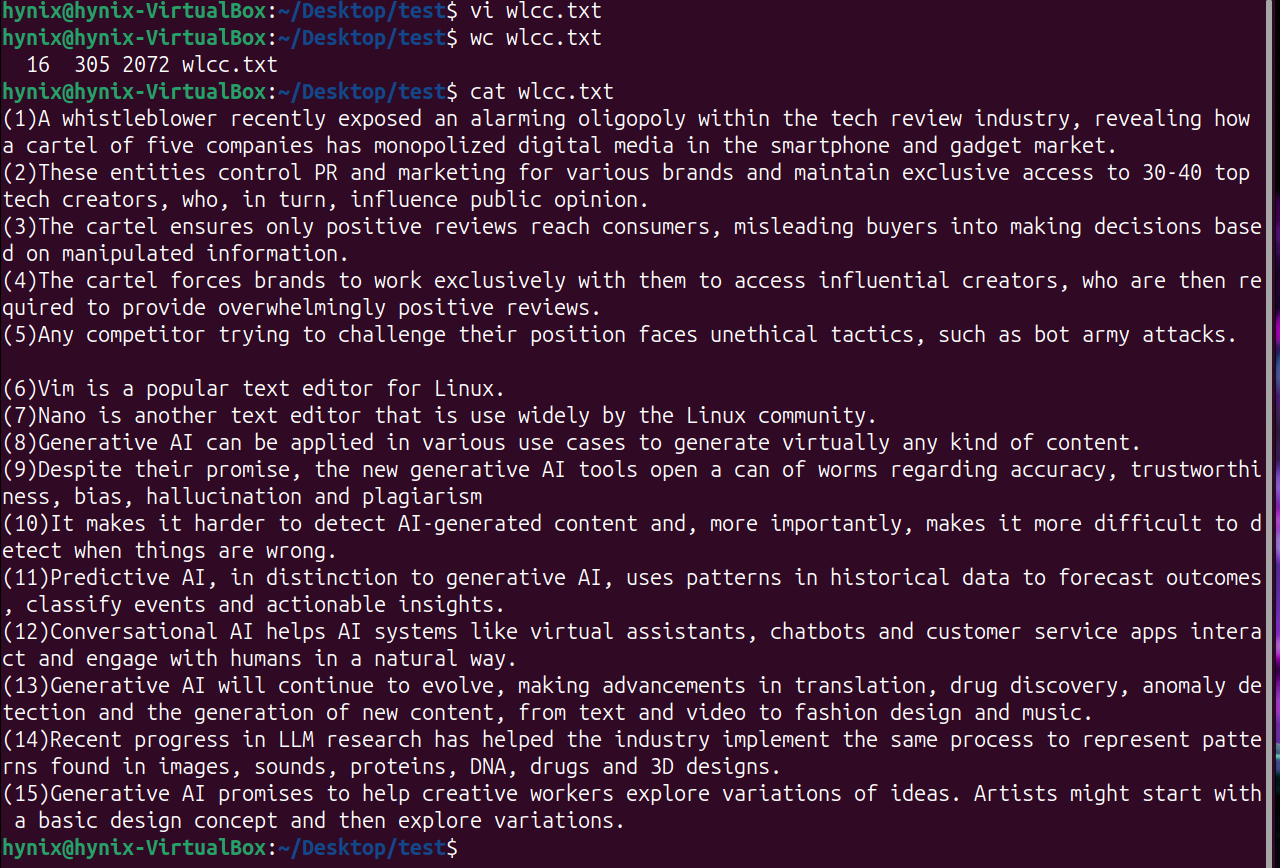
Contents of wlcc.txt -



Steps involved :-

1. Opened the file wlcc.txt in vi editor.
2. Then pressed G followed by o to start editing the file from 5th line onwards.
3. After adding those lines pressed ESC and type :wq to save the state of the file and quit.

Appended wlcc.txt -



# Experiment - 8

## Study and use of commands for changing file permissions

The chmod command is used to change the permissions of a file in Linux. The file permissions control the access to read, write and execute a file.

* + Read **(r)** – Permission to read the file or list the directory.
  + Write **(w)** – Permission to modify the file or create/delete files in a directory.
  + Execute **(x)** – Permission to execute the file or traverse a directory.

The basic syntax for **chmod** is:

chmod [options] [permissions] [file or directory]

### Operations:

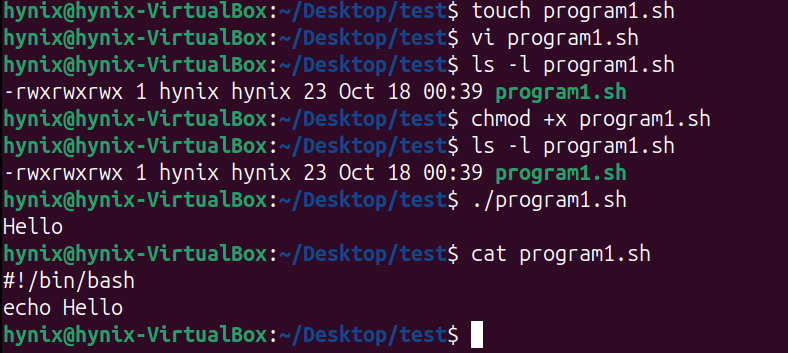
● **+** : Adds the specified permission.

* **-** : Removes the specified permission.

● = : Sets the exact permission (replacing previous permissions).

To view the current permissions of a file or directory, we use **ls -l** command.

I have used **chmod +x filename** to change the permissions of the file to executable in the below image.



**Symbolic Notation**: Uses characters to set file permissions:

* **u** (user/owner), **g** (group), **o** (others).
* **r** (read), **w** (write), **x** (execute).
* Example: chmod u+rwx, g+rx, o-r file → Sets owner to read/write/execute, group to read/execute, removes read for others.

**Octal Notation**: Uses numbers to set permissions:

* **4** = read, **2** = write, **1** = execute.
* Permissions are set as a 3-digit code: **owner-group-others**.
* Example: chmod 755 file → Owner (rwx), group (r-x), others (r-x).

**Comparison**:

* **Symbolic**: Human-readable, easy to understand.
* **Octal**: Faster, compact, commonly used by advanced users.