

Edit Text Files from Shell Prompt

Edit Files with Vim

The fundamental design principle of Linux is that it supports storage of the information and configuration settings in text-based files.

Vim is an improved version of the vi editor, which is distributed with Linux and UNIX systems. Vim is a highly configurable and efficient editor that provides split-screen editing, color formatting, and highlighting for editing text.

Benefits of the Vim Editor

When a system uses a text-only shell prompt, you should know how to use at least one text editor for editing files.

You can then edit text-based configuration files from a terminal window or remote logins through the ssh command or the Web Console. You also do not need access to a graphical desktop to edit files on a server, and that server might not need to run a graphical desktop environment.

The key reason to learn Vim is that it is almost always installed by default on a server for editing text-based files.

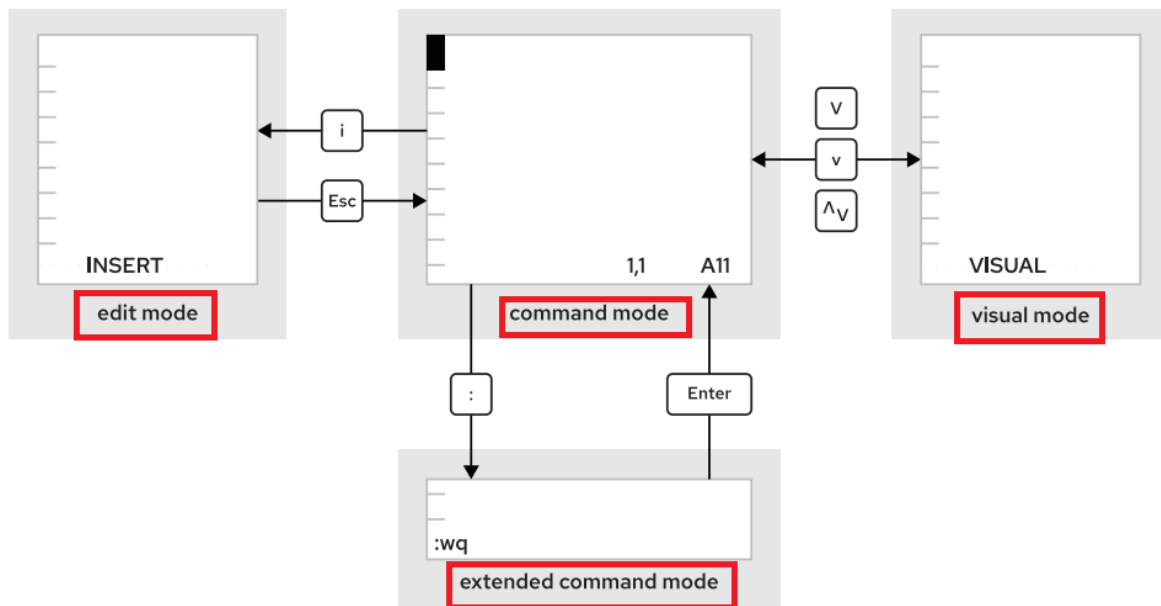
Vim is also used often as vi implementation on other standard operating systems or distributions. For example, macOS currently includes a lightweight installation of Vim by default. So, Vim skills that are learned for Linux might also prove useful elsewhere.

Get Started with Vim

- vim-minimal package → vi
- vi-enhanced → vim

```
[root@serverb ~]# rpm -qa | grep vim
vim-minimal-8.2.2637-20.el9_1.x86_64
vim-filesystem-8.2.2637-20.el9_1.noarch
vim-common-8.2.2637-20.el9_1.x86_64
vim-enhanced-8.2.2637-20.el9_1.x86_64
```

Vim Operating Modes



When you first open Vim, it starts in command mode,

- ➔ An “i” keystroke enters **insert mode**
- ➔ A “v” keystroke enters **visual mode**
- ➔ The : keystroke begins **extended command mode**

Command Mode:

Command Mode (also called **Normal Mode**) is the default mode when you open a file. In this mode, you can navigate, edit, and manipulate text using various commands.

Entering Command Mode

Press `Esc` if you're in **Insert Mode** or another mode.

Common Command Mode Operations

Navigation

- ➔ `h` → Move left
- ➔ `l` → Move right
- ➔ `j` → Move down
- ➔ `k` → Move up
- ➔ `w` → Move to the next word

www.linkedin.com/in/ragasudha-m

- ⌘ b → Move to the previous word
- ⌘ 0 → Move to the beginning of the line
- ⌘ ^ → Move to the first non-blank character of the line
- ⌘ \$ → Move to the end of the line
- ⌘ gg → Go to the beginning of the file
- ⌘ G → Go to the end of the file
- ⌘ Ctrl + d → Move half a screen down
- ⌘ Ctrl + u → Move half a screen up

Editing & Deleting

- ⌘ x → Delete the character under the cursor
- ⌘ dd → Delete (cut) the entire line
- ⌘ yy → Copy (yank) the entire line
- ⌘ p → Paste after the cursor
- ⌘ P → Paste before the cursor
- ⌘ u → Undo
- ⌘ Ctrl + r → Redo
- ⌘ r<char> → Replace the character under the cursor with <char>
- ⌘ cw → Change (replace) a word

Searching

- ✓ /word → Search for "**word**" in the file (press n for next match)

Edit Mode

In **Edit Mode**, you can directly type and modify text in a file. To enter **Edit Mode (Insert Mode)** from **Command Mode**, use one of the following commands:

Command	Action
i	Insert before the cursor
I	Insert at the beginning of the line
a	Append after the cursor
A	Append at the end of the line
o	Open a new line below the current line
O	Open a new line above the current line
cw	Change (replace) a word
C	Change the rest of the line
s	Delete the character under the cursor and start inserting

Visual Mode

Visual Mode allows you to select and manipulate text more efficiently. You can enter it from **Command Mode** and use various commands to modify the selected text.

Types of Visual Mode & How to Enter

Command	Mode	Description
v	Visual Mode	Selects text character by character
V	Visual Line Mode	Selects entire lines
Ctrl + v	Visual Block Mode	Selects a rectangular block of text

Common Operations in Visual Mode

Once you've selected text, you can perform various operations:

www.linkedin.com/in/ragasudha-m

Command	Action
y	Yank (copy) the selected text
d	Delete (cut) the selected text
x	Delete selected characters
p	Paste after the selection
P	Paste before the selection
>	Indent the selected text to the right
<	Indent the selected text to the left
~	Toggle case (uppercase/lowercase)
gU	Convert selected text to uppercase
gu	Convert selected text to lowercase
:	Run a command on the selected text (e.g., <code>: '<','>s/old/new/g</code> to replace text)

Extended Command Mode

Extended Command Mode, also known as **Ex Mode**, allows you to execute advanced commands, such as saving files, searching, and performing bulk edits.

Entering Ex Mode

- In **Command Mode**, press `:` (colon).
- The cursor moves to the bottom of the screen, where you can type commands.

Search & Replace in Ex Mode

Command	Description
<code>:s/old/new/</code>	Replace first occurrence of "old" with "new" on the current line
<code>:s/old/new/g</code>	Replace all occurrences of "old" with "new" on the current line
<code>:%s/old/new/g</code>	Replace all occurrences of "old" with "new" in the file
<code>:%s/old/new/gc</code>	Replace all occurrences but ask for confirmation

www.linkedin.com/in/ragasudha-m

Common Ex command Mode

Command	Description
<code>:w</code>	Save the file
<code>:q</code>	Quit Vim
<code>:wq</code> OR <code>:x</code>	Save and quit
<code>:q!</code>	Quit without saving
<code>:e filename</code>	Open another file
<code>:sav filename</code>	Save the file under a new name
<code>:set number</code>	Show line numbers
<code>:set nonumber</code>	Hide line numbers
<code>:set paste</code>	Enable paste mode (avoids auto-indentation issues)
<code>:set nopaste</code>	Disable paste mode
<code>:r filename</code>	Insert the contents of another file
<code>:!command</code>	Run a shell command (e.g., <code>:!ls</code> to list files)
<code>:w !sudo tee %</code>	Save the file with sudo (useful if you forgot to open it as root)

Note: #vimtutor is a command to learn about vim

Vim Configuration Files

- ➔ `/etc/vimrc` → Global Configuration File
- ➔ `~/.vimrc` → specific user

www.linkedin.com/in/ragasudha-m