Day-3-Vim Editor

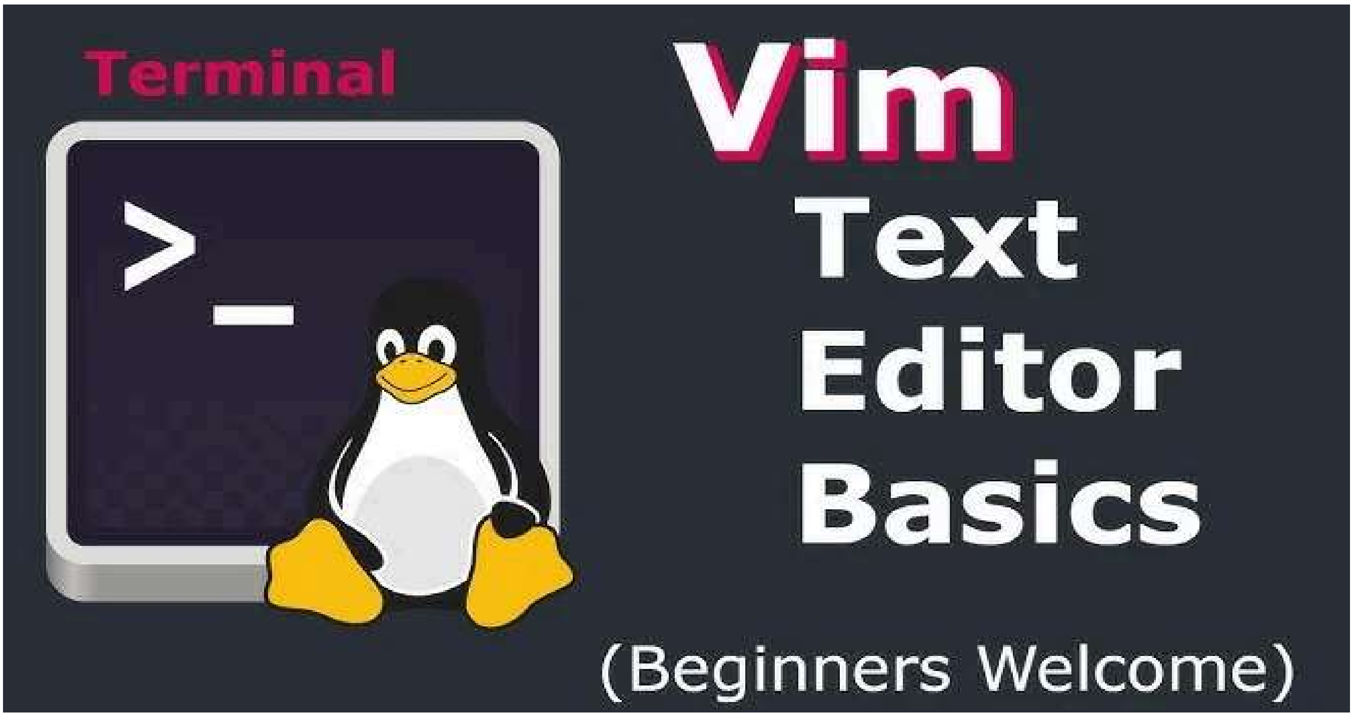
Dashboard

Day-3-Vim Editor and creating links

Step-by-Step Vim Guide a for Beginners and Hardlink softlink explained

mounika pogakula

May 17, 2025 · 6 min read



10

Table of co

Show less

The `vim` editor (Vi IMproved) is a powerful, flexible text editor commonly used in Linux and Unix systems. It’s widely used by system administrators, developers, and DevOps engineers because it's lightweight and available by default on most Unix systems.

🟩 What is Vim?

Vim stands for **Vi Improved**. It’s a powerful **text editor** available on almost all Unix-based systems (like Linux).

It's used to:

- Edit configuration files

- Write shell scripts
- View and manipulate text files

VIM EDITOR
VI Visual display editor
VIM Visual display editor improved

This is command mode editor for files. Other editors in Linux are emacs, gedit
vi editor is most popular

It has 3 modes:
1 **Command Mode**
2 **Insert mode (edit mode)**
3 **Extended command mode**

Note: When you open the vim editor, it will be in the command mode by default.

In the command mode the cursor's can be used as
h/j/k/l to move cursor left/right/up/down

🟩 How to Open Vim?

```
vim filename
```

Examples:

```
vim myfile.txt
```

- If `myfile.txt` doesn't exist, it creates a new file.
- If it exists, V
- **Step 2: Enter Insert Mode**

- Press `i`
  → You'll see `-- INSERT --` at the bottom.
- Now you can type anything:

COPY

```
Hello, this is a test file.
Learning Vim is fun!
```

Step 3: Exit Insert Mode

- Press `Esc`
  → You're now back in **Normal Mode** (no `-- INSERT --`).

Step 4: Save and Exit

- Type `:wq`
  → This **saves** the file and **quits** Vim.

🔁 Repeat:

- `i` → insert
- `Esc` → stop typing
- `:wq` → save and quit

Insert Mode:

Key	Description
i	position
I	To insert at the beginning of line

Key	Description
a	To append to the next word's letter
A	To Append at the end of the line
o	To insert a new line below the cursor position
O	To insert a new line above the cursor position

Command Mode:

Key	Description
gg	To go to the beginning of the page
G	To go to end of the page
w	To move the cursor forward, word by word
b	To move the cursor backward, word by word
nw	To move the cursor forward n words (5W)
nb	To move the cursor backward n to words (5B)
u	To undo last change (word)

Command Mode (continued):

Key	Description
U	To undo the previous changes (entire line)
Ctrl+R	To redo the changes
YY	
nyy	To copy n lines (5yy or 4yy)

Key	Description
p	To paste line below the cursor position
P	To paste line above the cursor position
dw	To delete the word letter by letter (like Backspace)
x	To delete the word letter by letter (like DEL Key)
dd	To delete entire line
ndd	To delete n no. of lines from cursor position (5dd)
/	To search a word in the file

Extended Mode : (Colon Mode)

Extended Mode is used for save and quit or save without quit using **“Esc”** key with **“:”**

Command	Description
Esc+:w	To Save the changes
Esc+:q	To quit (without saving)
Esc+:wq	To save and quit
Esc+:w!	To save forcefully
Esc+:wq!	To save and quit forcefully
Esc+:x	To save and quit
Esc+:X	To give password to the file and remove password
Esc+:20,10 n	To go to line no 20 or n
Esc+: se nu	
Esc+: se nonu	To Remove the set line numbers

Summary:

- `vim filename` – Open a file
- `i` – Insert mode to start typing
- `Esc` – Exit insert mode
- `:wq` – Save and exit

Symbolic Link

What is a Link?

A **link** is like a shortcut or reference to a file. It allows you to access a file from different locations without duplicating the file's content.

There are two types of Links:

	Soft Link	Hard Link
1	Size of link file is equal to no. of characters in the name of original file	Size of both file is same
2	Can be created across the Partition	Can’t be created across the partition
3	Inode no. of source and link file is different	Inode no. of both file is same
4	If original file is deleted, link is broken and data is lost	If original file is deleted then also link will contain data
5	SHORTCUT FILE	BACKUP FILE

Creating a soft link:

📝 Exercise 1: Create and Explore a Soft Link

1. **Create a test file:**

   COPY

   ```
   echo "This is the original file." > original.txt
   ```

2. **Create a soft link named** `softlink.txt` **:**

   COPY

   ```
   ln -s original.txt softlink.txt
   ```

3. **Check the details of the soft link:**

   COPY

   ```
   ls -l softlink.txt
   ```

- It shows `softlink.txt -> original.txt`

4. **Read the soft link file:**

   COPY

   ```
   cat softlink.txt
   ```

- You see the content of the original file.

5. **Delete the original file:**

   COPY

   ```
   rm original.tx
   ```

6. **Try reading the soft link again:**

COPY

```
cat softlink.txt
```

- You will get an error like `No such file or directory` because the link is broken.

Creating a hard link:

📝 Exercise 1: Create and Explore a Hard Link

1. **Create a test file:**

COPY

```
echo "This is the original file." > original.txt
```

2. **Create a hard link named** `hardlink.txt`**:**

COPY

```
ln original.txt hardlink.txt
```

3. **Check the inode numbers of both files:**

COPY

```
ls -i original.txt hardlink.txt
```

- You should see the **same inode number** for both files.

4. **Modify the hard link file:**

COPY

```
echo "Adding a line via hard link." >> hardlink.txt
```

5. **Check contents of the original file:**

COPY

```
cat original.txt
```

- You will see the added line, because both are the same file on disk.

6. **Delete the original file:**

COPY

```
rm original.txt
```

7. **Check the content of the hard link file:**

COPY

```
cat hardlink.txt
```

- The data is still accessible since the hard link points directly to the inode.

Summary

- Hard links share the same data and inode; deleting one doesn't remove data until all hard links are deleted.
- Soft links point to the filename; deleting the original file breaks the link.

vim linux · vi editor · linux for beginners · linux-basics · linux-commands · Linux · #HardLinks #SoftLinks · #linux#linux Commands#Devops90dayschallenge#task5