

MODULE 4 :

vi Editor, File & Directory Commands



The vi Editor

- Has advanced and powerful features
- Not intuitive
- Harder to learn than nano
- Requires time investment



The vi Editor

`vi [file]`

Edit file.

`vim [file]`

Same as vi, but more features.

`view [file]`

Starts vim in read-only mode.



vi Command Mode and Navigation

k

Up one line.

j

Down one line.

h

Left one character.

l

Right one character.

w

Right one word.

b

Left one word.

^

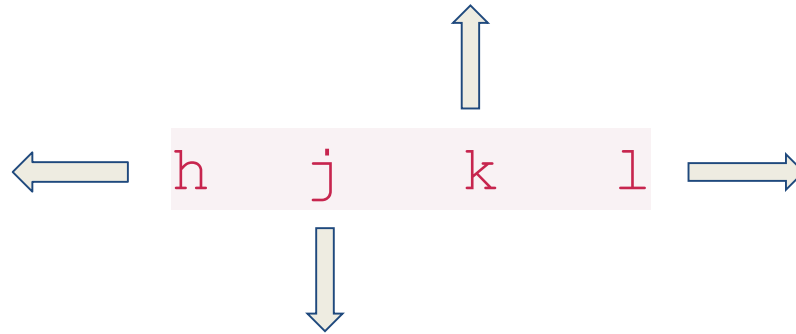
Go to the beginning of the line.

\$

Go to the end of the line.



vi Navigation Keys



vi Insert Mode

i

Insert at the cursor position.

I

Insert at the beginning of the line.

a

Append after the cursor position.

A

Append at the end of the line.



vi Line Mode

`:w`

Writes (saves) the file.

`:w!`

Forces the file to be saved.

`:q`

Quit.

`:q!`

Quit without saving changes.

`:wq!`

Write and quit.

`:x`

Same as :wq.



vi Line Mode(contd.)

`:n`

Positions the cursor at line n.

`:$`

Positions the cursor on the last line.

`:set nu`

Turn on line numbering.

`:set nonu`

Turn off line numbering.

`:help [subcommand]`

Get help.

vi Modes

Mode

Command

Insert

Line

Key

Esc

i I a A

:



vi - Repeating Commands

Repeat a command by preceding it with a number.

- `5k` = Move up a line 5 times
- `80i<Text><ESC>` = Insert `<Text>` 80 times
- `80i_<ESC>` = Insert 80 `"_"` characters

vi - Deleting Text

x

Delete a character.

dw

Delete a word.

dd

Delete a line.

D

Delete from the current position.



vi - Changing Text

r

Replace the current character.

cw

Change the current word.

cc

Change the current line.

c\$

Change the text from the current position.

C

Same as c\$.

~

Reverses the case of a character.



vi - Copying and Pasting

`yy`

Yank (copy) the current line.

`y<position>`

Yank the <position>.

`p`

Paste the most recent deleted or yanked text.

vi - Undo / Redo

u

Undo

Ctrl-R

Redo



vi - Searching

`/<pattern>`

Start a forward search.

`?<pattern>`

Start a reverse search.



vi - Repeating the Last Editor Command

You can just use "."

Example: You have "abc" at 10 places in your file and you want to replace it with "def" at 5 places of it.

Step 1: Find first occurrence of abc by typing command `/abc`

Step 2: Once cursor is on "abc", Replace abc by command `cw` to take out word "abc"

vi - Repeating the Last Editor Command (contd.)

Step 3: Type in "**def**" as replacement and press **ESC** to go to command mode

Step 4: To repeat this action just type command "**n**" to go to next occurrence of abc and **type command "."**

The command remembers that you replaced "**abc**" with "**def**" last time and will perform the same here.

vi - Joining lines

- pressing "J" at any place in the line you can combine the current line and the next line in the way you want.
- upper case "J" joins lines, lower case moves cursor.

vi - Writing Selected Lines onto a Separate File

If you want to create a new file with specific lines from existing file, use the following in command mode:

```
:1,5w <filename>
```

This will copy the first 5 lines from existing file to a new file.

If you want to overwrite the newly created file append the line numbers with **!**

```
:1,5w! <filename>
```

vi - Recovering files in vi editor

If you were disconnected from a **Unix** system while in an editing session using **vi**, you can still retrieve your document.

When the system detects the loss of your connection, it will retain a copy of your document as a hidden *swap* file similar to the following example:

```
Vim: Caught deadly signal TERM
Vim: preserving files...

Vim: Finished.
```

vi - Recovering files in vi editor(contd.)

To recover the file, use the command: `vi -r filename`

Using swap file ".file1.txt.swp"

Original file "/home/akhan/Documents/demo/important/file1.txt"

Recovery completed. You should check if everything is OK.

(You might want to write out this file under another name and run diff with the original file to check for changes)

You may want to delete the .swp file now.

Press ENTER or type command to continue

vi - Using the Shell from vi

If you want to execute a Unix command from inside the vi editor.

Step1: Go to command mode by pressing **ESC** key

Step2: Use **!** and type the name of the command

Example: **!: date**

Step3: Press ENTER to return to your editing session.

Thank You!