

Vi for nano users

markus schnalke <meillo@marmaro.de>
(Inspired and motivated by Toni Mueller)

Motivation

Vi is a powerful editor and hence frequently used by professionals. Novices often use nano which is known to be a “small and friendly” text editor. This document wants to show that learning the basics of vi is not much harder than learning nano, especially if vim is used.

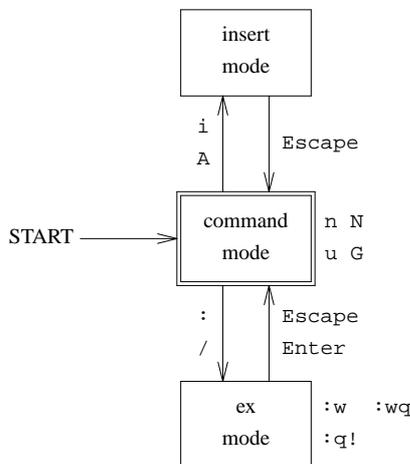
The modal interface

Learning vi means understanding its modal interface which is its basic concept. It makes vi “strange” but powerful. Once having understood the modal interface the rest will be easy.

When you startup vi you’ll always find yourself in the **command mode**, the central mode. Hitting the ‘Escape’ key brings you always back to the command mode.

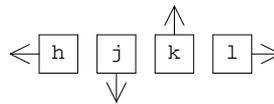
To insert text you need to switch to **insert mode**. You can do this by hitting ‘i’ (insert before cursor) or ‘A’ (append to end of line). When you have finished editing, ‘Escape’ to command mode.

The **ex mode** covers administrative operations. You enter it by typing ‘:’. See the colon appearing on the last terminal line. ‘:w’ writes the file out, ‘:wq’ writes and quits, ‘:q!’ quits without saving. ‘Enter’ executes the operation, ‘Escape’ aborts it. In both cases you return to command mode (or to the shell in case of quit).



Cursor movement

This is easy if you use vim: The cursor keys and PgUp/PgDn work both in command and insert mode. With vi you can only move the cursor in command mode with: ‘h’ (left), ‘j’ (down), ‘k’ (up), ‘l’ (right), ‘Ctrl-f’ (page forward), and ‘Ctrl-b’ (page backward).



Additional important functions

To search for text (regular expressions, actually) hit ‘/’ in command mode, insert a pattern, then hit ‘Enter’. This moves the cursor to the next match. You can jump from match to match with ‘n’ (forward) and ‘N’ (backward).

To jump to a specific line type the line number (it’s not printed anywhere) and ‘G’ in command mode. ‘1G’ means the first line, ‘G’ without number means the last line.

To undo the last change hit ‘u’. It depends on your implementation if you have multiple undo steps or just one.

Why vi and not simply nano?

Because vi is omnipresent and because vi scales well when you advance.

Edit files more efficient! With vi you can incrementally learn further functions and shortcuts that improve your productivity.

This document shows only how to access the vi features that typical nano users usually use. Using vi on this level is hardly better than using nano, but it is the basis for mastering vi.

The command mode offers a lot of functions you’ll never want to miss once you’ve learnt them. Professional users are able to do really impressive operations in ex mode, too. Actually, vi is two powerful editors in one: ex and vi.

Learn vi, you won’t regret it!