

1 Giới thiệu chung

- Vim là tên viết tắt của Vi IMproved, một editor giống như Vi với nhiều cải tiến.
- Vi là một editor “kinh điển” trên UNIX (và khó rất ưa).
- Đa số các phiên bản Linux hiện nay đều dùng Vim thay cho Vi; ngoại lệ duy nhất là Debian. Muốn có Vim trên Debian ta phải cài gói **vim**; khi đó **vi** sẽ là symlink đến **vim**. Muốn biết ta đang dùng Vim hay Vi thì trong Vi ta gõ **:ver**
- Bài viết này giả sử bạn đã sử dụng Vi vài lần và biết các thao tác cơ bản trong Vi (như khởi động, thoát, chèn, xóa) để có thể soạn thảo một tập tin đơn giản; đã biết các khái niệm Normal mode, Insert/Replace mode, Command-line mode, substitution.
- Bài viết này giới thiệu cách dùng một số tính năng có ích trong Vim để thực hiện các thao tác thường gặp một cách hiệu quả. Bài viết này không nhắm tới những người dùng tương đối thành thạo, mà chỉ giới thiệu những ví dụ đơn giản và hay gặp đối với người mới dùng Vim.

2 Sử dụng Vim một cách có hiệu quả

2.1 Khởi động

- các tùy chọn (options) của người dùng nên lưu trong **\$HOME/.vimrc**
- trong **.vimrc** nên có ít nhất các dòng sau:

```
set nosp
set hidden
set bs=2
syntax on
```

set nosp – “bật” các cải tiến của Vim

set hidden – cho phép soạn thảo nhiều file cùng một lúc trên một cửa sổ

set bs=2 – cho phép dùng BackSpace một cách “thân thiện” hơn

syntax on – thêm màu mè cho vui mắt

- Ghi nhớ: những thao tác nào hay lặp lại nhiều lần, trong đa số trường hợp có thể tự động hóa. Để sử dụng Vim cho có hiệu quả, nên:
 1. làm biếng gõ: cố tránh việc làm đi làm lại một thao tác nhiều lần;
 2. làm siêng đọc:
 - (a) thỉnh thoảng đọc lướt qua các mẹo dùng Vim do những người dùng Vim đã lâu mách. Quick start: tìm "**vim tips**" trên Google.
 - (b) Tập dùng on-line help của Vim; ví dụ **:help** ghi hướng dẫn chung về Vim, **:help nosp** để xem **nosp** làm gì, etc.
 - (c) Xem qua các mẹo trong Vim on-line manual: gõ **:help tips**

2.2 Sử dụng Undo/Redo

- sau khi lỡ tay quậy lung tung, nếu có hồi hận thì gõ `u` (Undo) trong Normal mode. Có thể gõ `u` nhiều lần.
- Gõ `u` một hồi lại hồi hận, muốn quay lại tình trạng trước đó thì gõ `<Ctrl-R>` (Redo) một hay nhiều lần.
- Có thể dùng Undo hay Redo ngay cả sau khi ta đã ghi file lên đĩa. Nhưng sau khi thoát ra khỏi Vim hoặc khi đã dùng `:e!` thì không còn Undo/Redo được nữa.

2.3 Sử dụng Visual mode

- copy/cut/paste một số dòng: trong Normal mode gõ `V`, di chuyển con trỏ (dùng các mũi tên hoặc `h`, `j`, `k`, `l`) để đánh dấu các dòng cần thao tác. Sau đó dùng:
 - `y` – ghi nhớ các dòng được tô sáng vào 1 register (copy)
 - `d` – như trên, đồng thời xóa các dòng này khỏi file (cut)
 - `p` – chèn các dòng trong register vào vị trí con trỏ (paste)
- sau khi tô sáng một số dòng, có thể thực hiện các Command-line commands chỉ trên đoạn tô sáng. Ví dụ:
 - chèn ký tự `#` vào đầu các dòng đang được tô sáng: gõ `:s/^/#`
 - xóa ký tự `#` ở đầu các dòng đang được tô sáng: gõ `:s/^#/`
 - chèn ngày tháng hiện tại vào file đang soạn thảo: di trỏ đến 1 dòng rỗng, gõ `V:!date`
- có thể đánh dấu 1 block thay vì một số dòng như trên: dùng `<Ctrl-V>` thay cho `V`.

2.4 Sử dụng auto-completion

- Trong insert/replace mode: ví dụ gõ `abc<Ctrl-P>` hoặc `abc<Ctrl-N>`, Vim sẽ tự động ghi tiếp phần còn lại nếu trong buffer đã có ít nhất 1 từ bắt đầu bằng `abc`. Gõ tiếp `<Ctrl-P>` để tìm các từ tiếp theo. Nhanh và đỡ sai sót khi viết các tên dài và khó nhớ.
- Filename completion trong Command-line mode: các thao tác có liên quan đến tên file, khi gõ phần đầu của tên file rồi ấn `<Tab>`, Vim sẽ tự động điền phần còn lại
- các tùy chọn (options): khi gõ ví dụ `:set a<Tab>`, Vim sẽ liệt kê tất cả các lựa chọn bắt đầu bằng `a` cho ta chọn.

2.5 Sử dụng command-line history

- sau khi gõ 1 lệnh substitution dài ngoằng mà nó không chạy do có lỗi, chán quá phải gõ lại rồi hử. Không sao, Vim nhớ những lệnh ta đã gõ trước đó, chỉ cần gõ lại 1,2 ký tự đầu rồi dùng `<Up>`, `<Down>` để chọn trong những lệnh mà ta đã gõ trước đó.
- Có thể sử dụng history cho các lệnh bắt đầu bằng `:`, `/` và `?`

2.6 Sử dụng multiple buffers/windows

Trong Vim ta có thể mở nhiều cửa sổ và nhiều file. Mỗi file sau khi đã mở tương ứng với 1 buffer. Ta có thể chuyển đổi giữa các cửa sổ, và trong 1 cửa sổ ta có thể chuyển đổi giữa các buffer. Nhớ dùng auto-completion khi cần gõ tên 1 file hay buffer.

2.6.1 Các lệnh liên quan đến cửa sổ

- `<Ctrl-W>S` – mở thêm 1 cửa sổ mới
- `<Ctrl-W>C` – đóng cửa sổ chứa con trỏ
- `<Ctrl-W>O` – đóng tất cả các cửa sổ khác trừ cửa sổ chứa con trỏ
- `:sn <filename>` – mở file `<filename>` trong 1 cửa sổ mới

2.6.2 Các lệnh liên quan đến buffers

- `:e <filename>` – soạn thảo 1 file mới trong cửa sổ hiện thời
- `:b <filename>` – chuyển sang file `filename` trong cửa sổ hiện thời
- `:sn <filename>` – mở file `<filename>` trong 1 cửa sổ mới
- `:sb <filename>` – mở buffer `<filename>` trong 1 cửa sổ mới
- `:ls` – liệt kê các buffer đang mở
- `:wa` – ghi tất cả các buffer đang mở lên đĩa.

2.7 Gõ tiếng Việt trong Vim

Xem <http://vinux.sf.net/vim-vn>

2.8 Hãy cho tôi thấy .vimrc của bạn, tôi sẽ cho bạn biết ...

Xem <http://vinux.sf.net/vim-intro/myvimfiles.tgz>

3 Góp ý

Xin gửi đến VietLUG mailing list <mailto:vietylug-users@lists.sourceforge.net> hoặc đến tác giả <mailto:hanthethanh@gmx.net>