

第九章 VIM 程序编辑器

一、VI 的使用

1. 关于 vi

vi 共分为三种模式，分别是一般命令模式、编辑模式与命令行模式。这三种模式的作用分别是：

(1) 一般模式：以 vi 打开一个文件就直接进入一般模式了。在这个模式中，你可以使用『上下左右』按键来移动光标，你可以使用『删除字符』或『删除整行』来处理文件内容，也可以使用『复制、粘贴』来处理你的文件数据。

(2) 编辑模式：在一般模式中可以进行删除、复制、粘贴等等的动作，但是却无法编辑文件内容。按下“i,I,o,O,a,A,r,R”等任何一个字母之后才会进入编辑模式。通常在 Linux 中，按下这些按键时，在画面的左下方会出现『INSERT 或 REPLACE』的字样，此时才可以进行编辑。而如果要回到一般模式时，则必须要按下『Esc』这个按键即可退出编辑模式。

(3) 命令行模式：在一般模式中，输入“:/?”三个中的任何一个，就可以将光标移动到最下面那一行。在这个模式之中，可以搜索数据，也可以读取、保存、大量替换字符、离开 vi、显示行号等。

总的来说，这三种模式的关系如图所示：

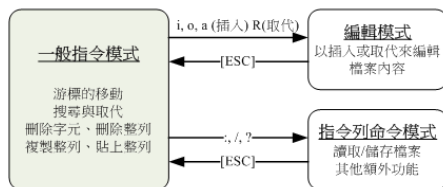


图 1.1 三种模式关系图

2. 按键说明

第一部分：一般模式可以使用的命令：光标移动、复制粘贴、查找替换等。

移动光标的方法	
h 或向左箭头键(←)	光标向左移动一个字符
j 或向下箭头键(↓)	光标向下移动一个字符
k 或向上箭头键(↑)	光标向上移动一个字符
l 或向右箭头键(→)	光标向右移动一个字符
[Ctrl]+[f]	屏幕『向下』移动一页，相当于[PageDown]按键（常用）
[Ctrl]+[b]	屏幕『向上』移动一页，相当于[PageUp]按键（常用）
[Ctrl]+[d]	屏幕『向下』移动半页
[Ctrl]+[u]	屏幕『向上』移动半页
+	光标移动到非空格符的下一行
-	光标移动到非空格符的上一行
n<space>	n 表示数字。按下数字后再按空格键，会向右移动 n 个字符。
0 或功能键[Home]	这是数字 0：移动到这一行的最前面字符处（常用）
\$或功能键[End]	移动到这一行的最后面字符处（常用）
H	光标移动到这个屏幕的最上方那一行的第一个字符
M	光标移动到这个屏幕的中央那一行的第一个字符
L	光标移动到这个屏幕的最下方那一行的第一个字符
G	移动到这个文件的最后一行（常用）

nG	n 为数字。移动到这个文件的第 n 行。
gg	移动到这个文件的第一行，相当于 1G 啊！（常用）
n<Enter>	n 为数字。光标向下移动 n 行（常用）
查找与替换	
/word	向光标之下寻找一个名称为 word 的字符串。
?word	向光标之上寻找一个字符串名称为 word 的字符串。
n	这个 n 是英文按键。代表『重复前一个搜寻的动作』。
N	这个 N 是英文按键。为『反向』进行前一个搜寻动作。
:n1,n2s/word1/word2/g	n1 与 n2 为数字。在第 n1 与 n2 行之间寻找 word1 这个字符串，并将该字符串替换为 word2。
:1,\$s/word1/word2/g	从第一行到最后一行寻找 word1 字符串，并将该字符串替换为 word2（常用）
:1,\$s/word1/word2/gc	从第一行到最后一行寻找 word1 字符串，并将该字符串替换为 word2！且在替换前显示提示字符给用户确认(confirm)是否需要替换（常用）
删除、复制与贴上	
x,X	在一行字当中，x 为向后删除一个字符(相当于[del]按键)，X 为向前删除一个字符(相当于[backspace]亦即是退格键)（常用）
nx	n 为数字，连续向后删除 n 个字符。举例来说，我要连续删除 10 个字符，『10x』。
dd	删除光标所在的那一整行（常用）
ndd	n 为数字。删除光标所在的向下 n 行，（常用）
d1G	删除光标所在到第一行的所有数据
dG	删除光标所在到最后一行的所有数据
d\$	删除光标所在处，到该行的最后一个字符
d0	删除光标所在处，到该行的最前面一个字符
yy	复制光标所在的那一行（常用）
nyy	n 为数字。复制光标所在的向下 n 行，（常用）
y1G	复制光标所在行到第一行的所有数据
yG	复制光标所在行到最后一行的所有数据
y0	复制光标所在的那个字符到该行行首的所有数据
y\$	复制光标所在的那个字符到该行行尾的所有数据
p,P	p 为将已复制的数据在光标下一行粘贴，P 则粘贴在上一行。
J	将光标所在行与下一行的数据结合成同一行
c	重复删除多个数据，例如向下删除 10 行，[10cj]
u	复原前一个动作。（常用）
[Ctrl]+r	重做上一个动作。（常用）
.	意思是重复前一个动作的意思。如果你想要重复删除、重复贴上等等动作，按下小数点『.』就好了！（常用）

第二部分：一般模式切换到编辑模式的可用的按键说明：

进入插入或替换的编辑模式

i,I	进入插入模式(Insert mode): i 为从目前光标所在处插入, I 为在目前所在行的第一个非空格符处开始插入。(常用)
a,A	进入插入模式(Insert mode): a 为从目前光标所在的下一个字符处开始插入, A 为从光标所在行的最后一个字符处开始插入。(常用)
o,O	进入插入模式(Insert mode): o 为在目前光标所在的下一行处插入新的一行; O 为在目前光标所在处的上一行插入新的一行(常用)
r,R	进入替换模式(Replace mode): r 只会替换光标所在的那一个字符一次; R 会一直替换光标所在的文字, 直到按下 ESC 为止。(常用)
[Esc]	退出编辑模式, 回到一般模式中(常用)

第三部分: 一般模式切换到命令行模式的可用的按键说明:

命令行模式的储存、离开等指令	
:w	将编辑的数据写入硬盘文件中(常用)
:w!	若文件属性为“只读”时, 强制写入该文件。不过, 是否能写入与你对文件的权限有关。
:q	离开 vi(常用)
:q!	若曾修改过文件, 又不想储存, 使用!为强制离开不储存文件。
:wq	储存后离开, 若为:wq! 则为强制储存后离开(常用)
ZZ	若文件没有更动, 则不储存离开, 若文件已经被更动过, 则储存后退出。
:w[filename]	将编辑的数据储存成另一个文件(类似另存为)
:r[filename]	在编辑的数据中, 读入另一个文件的数据。即将 filename 这个文件内容加到光标所在行后面
:n1,n2w[filename]	将 n1 到 n2 的内容储存成 filename 这个文件。
:!command	暂时离开 vi 到指令行模式下执行 command 的显示结果。
vim 环境的变更	
:setnu	显示行号, 设定之后, 会在每一行的前缀显示该行的行号。
:setnonu	与 setnu 相反, 为取消行号。

3.vim 的保存文件、恢复与打开时的警告信息

在目前主要的文本编辑软件都会有“恢复”的功能, 即当你的系统因为某些原因而导致类似死机的情况时, 还可以透过某些特别的机制来让你将之前未储存的数据恢复回来。VIM 是通过“保存文件”来挽回文件。

当我们在使用 vim 编辑时, vim 会在被编辑的文件的目录下, 再建立一个名为 filename.swp 的文件。如果你的系统因为某些原因死机, 导致编辑的文件没有保存, 这个时候该文件就能发挥恢复功能。

当死机以后, 由于 vim 的工作被不正常的中断, 会导致保存文件无法通过正常流程结束, 所以暂存文件就不会消失, 而是继续保留下来。此时, 如果我们想继续编辑那个文件, 则会出现以下警告:

```
E325: ATTENTION  <==錯誤代碼
Found a swap file by the name ".man_db.conf.swp"  <==底下多行说明有暂存档的存在
    owned by: dmtsai    dated: Mon Jul  6 23:54:16 2015
    file name: /tmp/vitest/man_db.conf  <==这个暂存文件是哪个文件的暂存档
    modified: no
    user name: dmtsai    host name: study.centos.vbird
    process ID: 31851
```

While opening file "man_db.conf"

dated: Mon Jul 6 23:47:21 2015

在这个暂存文件警告的界面中，有六个可以使用的按键，分别如下：

- (1) [O]pen Read-Only: 以只读模式打开此文件。
- (2) [E]dit anyway: 以正常方式打开你要编辑的文件，并不会载入暂存文件的内容。
- (3) [R]ecover: 载入暂存文件的内容。当你恢复文件后储存并打算离开 vim 的时候，需要自行删除 swp 文件。
- (4) [D]elete it: 删除暂存文件。
- (5) [Q]uit: 离开 vim。
- (6) [A]bort: 忽略这个编辑行为。

二、VIM 的额外功能

如果你使用 vi 后，却看到画面的右下角有显示目前光标所在的行行号码，那么你的 vi 已经被 vim 所取代。

1. 块选择

当我们按下 v 或者 V 或者 [Ctrl]+V 时，这个时候光标移动过的地方就会开始反白，这三个按键的意义如下。

区块选择的按键意义	
v	字符选择，会将光标经过的地方反白选择。
V	行选择，会将光标经过的行反白选择。
[Ctrl]+v	区块选择，可以用长方形的方式选择资料。
y	将反白的地方复制。
d	将反白的地方删除。
p	将刚刚复制的区块，在光标所在处粘贴。

2. 多文件编辑

使用 vim 的按键同时打开多个文档：

多文件编辑的按键	
:n	编辑下一个文件
:N	编辑上一个文件
:files	列出目前这个 vim 的开启的所有文件

3. 多窗口功能

在命令列模式输入『:sp {filename}』即可！那个 filename 可有可无，如果想要在新窗口启动另一个档案，就加入档名，否则仅输入 :sp 时，出现的则是同一个档案在两个窗口。

常用功能如下：

多窗口情况下的按键功能	
:sp [filename]	开启一个新窗口，如果有加 filename，表示在新窗口开启一个新档案，否则表示两个窗口为同一个档案内容(同步显示)。
[ctrl]+w+j [ctrl]+w+↓	按键的按法是：先按下 [ctrl] 不放，再按下 w 后放开所有的按键，然后再按下 j (或向下箭头键)，则光标可移动到下方的窗口。
[ctrl]+w+k [ctrl]+w+↑	同上，不过光标移动到上面的窗口。

[ctrl]+w+q	其实就是 :q 结束离开啦！ 举例来说，如果我想要结束下方的窗口，那么利用 [ctrl]+w+↓ 移动到下方窗口后，按下 :q 即可离开， 也可以按下 [ctrl]+w+q 啊！
------------	---

4.vim 的字补全功能

组合按钮	补齐的内容
[ctrl]+x -> [ctrl]+n	透过目前正在编辑的这个『文件的内容文字』作为关键词，予以补齐
[ctrl]+x -> [ctrl]+f	以当前目录内的『文件名』作为关键词，予以补齐
[ctrl]+x -> [ctrl]+o	以扩展名作为语法补充，以 vim 内建的关键词，予以补齐

5.vim 环境设置与记录：~/.vimrc，~/.viminfo

vim 会主动的将你曾经做过的行为登录下来，那个记录动作的文件就是： ~/.viminfo
可以输入:set all 查看所有的选项，如图 2.1 所示。

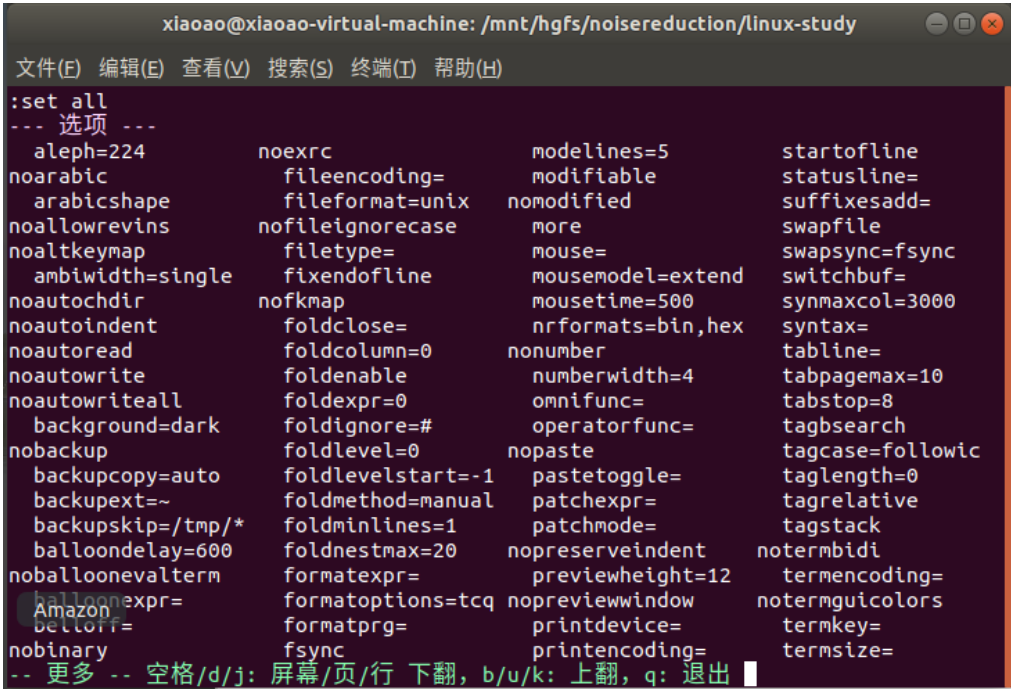


图 2.1 setall 选项图

各个选项设置如下：

vim 的环境设定参数	
:set nu :set nonu	设定与取消行号
:set hlsearch	hlsearch 就是 highlightsearch(高亮度搜寻)。
:set autoindent :set noautoindent	是否自动缩排。autoindent 就是自动缩排。
:set backup	是否自动储存备份档。一般是 nobackup 的。
:set ruler	就是在显示或不显示该设定值的啦！
:set showmode	是否要显示--INSERT--之类的字眼在左下角的状态栏。

:set backspace=(012)	一般来说，如果我们按下 i 进入编辑模式后，可以利用退格键 (backspace) 来删除任意字符的。但是，某些 distribution 则不许如此。此时，我们就可以透过 backspace 来设定啰～当 backspace 为 2 时，就是可以删除任意值；0 或 1 时，仅可删除刚刚输入的字符，而无法删除原本就已经存在的文字了！
:set all	显示目前所有的环境参数设定值。
:set	显示与系统默认值不同的设定参数，一般来说就是你有自行变动过的设定参数啦！
:syntax on :syntax off	是否依据程序相关语法显示不同颜色
:set bg=dark :set bg=light	可用以显示不同的颜色色调

当我们想要使用自己的习惯而不是 VIM 默认习惯的时候，可以修改 ~/.vimrc 这个文件。比如说你可以这样写：

```
[dmtsai@study ~]$ vim ~/.vimrc"这个档案的双引号 (") 是批注
set hlsearch          "高亮度反白
set backspace=2       "可随时用退格键删除
set autoindent        "自动缩排
```

6.vim 常用指令示意图



图 2.2 常用指令示意图（图片来源于 deniro）

三、其它 VIM 使用注意事项

1. 中文编码的问题

修正中文编码: LANG=zh_CN.UTF-8

2. DOS 与 Linux 的断行字符

在 DOS 使用的断行字符为^M\$, 我们称为 CR 与 LF 两个符号。而在 Linux 底下, 则是仅有 LF(\$)这个断行符号。

在 Linux 底下的命令在开始执行时, 他的判断依据是『Enter』, 而 Linux 的 Enter 为 LF 符号, 不过, 由于 DOS 的断行符号是 CRLF, 也就是多了一个^M 的符号出来, 在这样的情况下, 如果是一个 shell script 的程序档案, 呵呵~将可能造成『程序无法执行』的状态, 因为他会误判程序所下达的指令内容

我们所需要的指令默认没有安装, 在没有网络的情况下, 使用安装光盘, 然后输入下面的命令 (在 root 权限下)。

```
[root@study ~]# mount /dev/sr0 /mnt
[root@study ~]# rpm -ivh /mnt/Packages/dos2unix-*
warning: /mnt/Packages/dos2unix-6.0.3-4.el7.x86_64.rpm: Header V3 RSA/SHA256 ....
Preparing...                               ##### [100%]
Updating / installing...
   1:dos2unix-6.0.3-4.el7                    ##### [100%]
[root@study ~]# umount /mnt
[root@study ~]# exit
```

命令格式如下。

```
[dmtsai@study ~]$ dos2unix [-kn] file [newfile]
[dmtsai@study ~]$ unix2dos [-kn] file [newfile]
选项与参数:
-k   : 保留该文件原本的 mtime 时间格式 (不更新档案上次内容经过修订的时间)
-n   : 保留原本的旧档, 将转换后的内容输出到新档案, 如:  dos2unix -n old new
```

3. 语系编码转换

如何将 GBK 编码转成 UTF8 编码呢? 可以使用 iconv 指令。

```
[dmtsai@study ~]$ iconv --list
[dmtsai@study ~]$ iconv -f 原本编码 -t 新编码 filename [-o newfile]
选项与参数:
--list: 列出 iconv 支持的语系数据
-f: from, 亦即来源之意, 后接原本的编码格式;
-t: to, 亦即后来的新编码要是什么格式;
-o file: 如果要保留原本的档案, 那么使用-o 新文件名, 可以建立新编码文件。
```