

工具1-Vi(Vim)键盘图及其基本命令

Vi(Vim)键盘图及其基本命令:

version 1.1
April 1st, 06
翻译:2006-5-22

vi / vim 键盘图

Esc
命令模式

~ 转换大小写
跳转到标注

! 外部过滤器

@ 运行宏

prev ident

\$ 行末

% 括号匹配

^ "款" 行首

& 重复 :s

* next ident

(句首

) 下一句首

"soft" bold down

+ 后一行行首

1

2

3

4

5

6

7

8

9

0 "硬" 行首

- 前一行行首

= 自动格式化

Q 切换到 ex 模式

W 下一单词

E 词尾

R 替换模式

T back 'till

Y 拷贝行

U 撤销行内命令

I 到行首插入

O 分段(前)

P 粘贴(前)

{ 段首

} 段尾

q 录制宏

w 下一单词

e 词尾

r 替换字符

t 'till

y 拷贝

u 撤销命令

i 插入模式

o 分段(后)

p 粘贴(后)

· 杂项

· 杂项

A 在行末附加

S 删除行并插入

D 删除至行末

F 行内字符反向查找

G 文尾/行号

H 屏幕顶行

J 合并两行

K 帮助

L 屏幕底行

:

ex 命令

" 寄存器标识

' 跳到标注的行首

行首/列

a 附加

s 删除字符并插入

d 删除

f 行内字符查找

g 附加命令

h ←

j ↓

k ↑

l →

;

重复 :u/T/f/F

' 跳到标注的行首

\ 未使用!

Z 退出

X 退格

C 修改至行末

V 可视行模式

B 前一单词

N 查找上一处

M 屏幕中间行

< 反缩进

> 缩进

? 向前搜索

Z 附加命令

x 删除(字符)

c 修改

v 可视模式

b 前一单词

n 查找下一处

m 设置标注

, 反向 u/T/f/F

· 重复命令

/ 向后搜索

动作 移动光标, 或者定义操作的范围

命令 直接执行的命令, 红色命令进入编辑模式

操作 后面跟随表示操作范围的指令

extra 特殊功能, 需要额外的输入
q 后跟字符参数

主要ex命令:
:w (保存), :q (退出), :q! (不保存退出)
:e f (打开文件 f),
:%s/x/y/g ('y' 全局替换 'x'),
:h (帮助 in vim), :new (新建文件 in vim)

其它重要命令:
CTRL-R: 重复 (vim),
CTRL-F/-B: 上翻/下翻,
CTRL-E/-Y: 上滚/下滚,
CTRL-V: 块可视模式 (vim only)

可视模式:
漫游后对选中的区域执行操作 (vim only)

备注:
(1) 在拷贝/粘贴/删除命令前使用 "x (x=a..z,*) 使用命令的寄存器(剪贴板)"
(如: "ay8 拷贝剩余的行内容至寄存器 'a')
(2) 命令前添加数字 多遍重复操作
(e.g.: 2p, d2w, 5i, d4j)
(3) 重复本字符在光标所在行执行操作
(dd = 删除本行, >> = 行首缩进)
(4) ZZ 保存退出, ZQ 不保存退出
(5) zt: 移动光标所在行至屏幕顶端,
zb: 底端, zz: 中间
(6) gg: 文首 (vim only),
gt: 打开光标处的文件名 (vim only)

原图: www.viemu.com 翻译: fdl (linuxsir)

进入 vi

vi filename	打开或新建文件, 并将光标置于第一行首
vi +n filename	打开文件, 并将光标置于第 n 行首
vi + filename	打开文件, 并将光标置于最后一行首
vi -r filename	在上次正用 vi 编辑时发生系统崩溃, 恢复文件
vi file1....filen	打开多个文件, 依次编辑

vi的工作模式

命令行模式 (command mode): 控制屏幕光标的移动, 字符、字或行的删除, 移动复制某区段及进入Insert mode下, 或者到 last line mode。

插入模式 (Insert mode): 只有在 Insert mode 下, 才可以做文字输入, 按「ESC」键可回到命令行模式。

底行模式 (last line mode): 将文件保存或退出 vi, 也可以设置编辑环境, 如寻找字符串、列出行号等。

命令行模式

移动光标

k, j, h, l	功能分别等同于上、下、左、右箭头键
Ctrl+b	在文件中向上移动一页 (相当于 PageUp 键)
Ctrl+f	在文件中向下移动一页 (相当于 PageDown 键)
ctrl+u	屏幕往后移动半页
ctrl+d	屏幕往前移动半页
H	将光标移到屏幕的最上行 (Highest)

nH	将光标移到屏幕的第 n 行（如 2H ： 将光标移到屏幕的第 2 行）
M	将光标移到屏幕的中间（ Middle ）
L	将光标移到屏幕的最下行（ Lowest ）
nL	将光标移到屏幕的倒数第 n 行（如 3L ： 将光标移到屏幕的倒数第 3 行）
w	在指定行内右移光标，到下一个字的开头
e	在指定行内右移光标，到一个字的末尾
b	在指定行内左移光标，到前一个字的开头
0	数字 0 ，左移光标，到本行的开头
G	光标移动到文章的最后
nG	光标移动到文章的第 n 行（如 8G ： 移动到文章的第 8 行）
\$	右移光标，到本行的末尾
^	移动光标，到本行的第一个非空字符

替换和删除

rc	用 c 替换光标所指向的当前字符
nrc	用 c 替换光标所指向的前 n 个字符（如 5rc ： 用 c 替换光标所指向的前 5 个字符）
x	删除光标所在位置后面的一个字符
nx	删除光标所在位置后面的 n 个字符（如 3x ： 删除光标所在位置后面的 3 个字符）
X	大写的 X ，删除光标所在位置前面的一个字符
nX	删除光标所在位置前面的 n 个字符（如 3X ： 删除光标所在位置前面的 3 个字符）
dd	删除光标所在行，并去除空隙
ndd	从光标所在行开始删除 n 行内容，并去除空隙（如 3dd ： 删除 3 行内容，并去除空隙）

复制和粘贴

从正文中删除的内容（如字符、字或行）并没有真正丢失，而是被剪切并复制到了一个内存缓冲区中。用户可将其粘贴到正文中的指定位置。

p	小写字母 p ，将缓冲区的内容粘贴到光标的后面
P	大写字母 P ，将缓冲区的内容粘贴到光标的前面

如果缓冲区的内容是字符或字，直接粘贴在光标的前面或后面；如果缓冲区的内容为整行正文则粘贴在当前光标所在行的上一行或下一行。

有时需要复制一段正文到新位置，同时保留原有位置的内容。这种情况下，首先应当把指定内容复制（而不是剪切）到内存缓冲区。完成这一操作的命令是：

yy	复制当前行到内存缓冲区
nyy	复制 n 行内容到内存缓冲区（如 5yy ： 复制 5 行内容到内存缓冲区）

搜索字符串

/str1	正向搜索字符串 str1
n	继续搜索，找出 str1 字符串下次出现的位置
?str2	反向搜索字符串 str2

撤销和重复

u	撤消前一条命令的结果
.	重复最后一条修改正文的命令

文本选中

v	字符选中命令
V	行选中命令

插入模式

进入插入模式

i	在光标左侧输入正文
a	在光标右侧输入正文
o	在光标所在行的下一行增添新行
O	在光标所在行的上一行增添新行
I	在光标所在行的开头输入正文
A	在光标所在行的末尾输入正文

退出插入模式

ESC 键或组合键Ctrl+[

底行模式

在vi的底行模式下，可以使用复杂的命令。

退出命令

在命令模式下可以用 ZZ 命令退出 vi 编辑程序，该命令保存对正文所作的修改，覆盖原始文件。如果只需要退出编辑程序，而不打算保存编辑的内容，可用下面的命令：

:q	在未作修改的情况下退出
:q!	放弃所有修改，退出编辑程序

行号和文件保存

:n	将光标移到第 n 行
:set nu	显示行号
:set nonu	取消行号显示

底行模式下，可以规定命令操作的行号范围。数值用来指定绝对行号；字符“.”表示光标所在行的行号；字符“\$”表示正文最后一行的行号；简单的表达式，例如“.+5”表示当前行往下的第 5 行。例如：

:.+5	将光标移到当前行之后的第5行
:\$	将光标移到正文最后一行

在底行模式下，允许从文件中读取正文，或将正文写入文件。例如：

:w	将编辑的内容写入原始文件，用来保存编辑的中间结果
:wq	将编辑的内容写入原始文件并退出编辑程序（相当于 ZZ 命令）
:w file	将编辑的内容写入 file 文件，保持原有文件的内容不变
:a,bw file	将第 a 行至第 b 行的内容写入 file 文件（如:1,w file 将第 1 行至当前行写入 file 文件）
:r file	读取 file 文件的内容，插入当前光标所在行的后面
:f file	将当前文件重命名为 file

字符串搜索

:/str/	正向搜索，将光标移到下一个包含字符串 str 的行
:?str?	反向搜索，将光标移到上一个包含字符串 str 的行

正文替换

:s/str1/str2/	用字符串 str2 替换行中首次出现的字符串 str1
:s/str1/str2/g	用字符串 str2 替换行中所有出现的字符串 str1
.,\$ s/str1/str2/g	用字符串 str2 替换正文当前行到末尾所有出现的字符串 str1
:1,\$ s/str1/str2/g	用字符串 str2 替换正文中所有出现的字符串 str1
:g/str1/s//str2/g	功能同上

从上述替换命令可以看到：**g** 放在命令末尾，表示对搜索字符串的每次出现进行替换；不加 **g**，表示只对搜索字符串的首次出现进行替换；**g** 放在命令开头，表示对正文中所有包含搜索字符串的行进行替换操作。

删除正文

<code>:d</code>	删除光标所在行
<code>:3d</code>	删除第 3 行
<code>:\$d</code>	删除当前行至正文的末尾
<code>:/str1/,/str2/d</code>	删除从字符串 str1 到 str2 的所有行

恢复文件

vi 在编辑某个文件时，会另外生成一个临时文件，这个文件的名称通常以 **.** 开头，并以 **.swp** 结尾。**vi** 在正常退出时，该文件被删除，若意外退出，而没有保存文件的最新修改内容，则可以使用恢复命令，也可以在启动 **vi** 时利用 **-r** 选项。

<code>:recover</code>	恢复文件
-----------------------	------

选项设置

为控制不同的编辑功能，**vi** 提供了很多内部选项。利用 **:set** 命令可以设置选项。基本语法为：

<code>:set option</code>	设置选项 option
--------------------------	--------------------

常见的功能选项包括：

autoindent	设置该选项，则正文自动缩进
ignorecase	设置该选项，则忽略规则表达式中大小写字母的区别
number	设置该选项，则显示正文行号
ruler	设置该选项，则在屏幕底部显示光标所在行、列的位置
tabstop	设置按 Tab 键跳过的空格数。例如 <code>:set tabstop=n</code> ， n 默认值为 8
mk	将选项保存在当前目录的 .exrc 文件中

shell切换

在编辑正文时，利用 **vi** 底行模式下提供的 **shell** 切换命令，无须退出 **vi** 即可执行 **Linux** 命令

<code>:! command</code>	执行完 shell 命令 command 后回到 vi
-------------------------	--

另外，在命令模式下，键入 **K**，可命令 **vi** 查找光标所在单词的手册页，相当于运行 **man** 命令。

-----分割线 -----

把**VIM**打成一个简单实用的**IDE** <http://www.linuxidc.com/Linux/2011-06/37032.htm>

Vim学习指南 <http://www.linuxidc.com/Linux/2013-08/89096.htm>

快速学会 **V**编辑器 <http://www.linuxidc.com/Linux/2013-08/88586.htm>

强大的**Vim** 编辑器 <http://www.linuxidc.com/Linux/2013-07/87544.htm>

在**CentOS** 6.2上搭建**Vim**开发环境 <http://www.linuxidc.com/Linux/2013-07/87363.htm>

CentOS 5.4 安装高亮**Vim**编辑工具 <http://www.linuxidc.com/Linux/2013-06/86508.htm>

Vim技巧分享:C语言设置 <http://www.linuxidc.com/Linux/2012-12/77124.htm>

Ubuntu中设置**Vim**的行号 <http://www.linuxidc.com/Linux/2012-12/75485.htm>

Vim编辑器使用基础教程 <http://www.linuxidc.com/Linux/2013-05/84031.htm>