



链滴

vi/vim 编辑工具

作者: [Smiteli](#)

原文链接: <https://ld246.com/article/1538101975055>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

一、vi/vim是Linux最基本的编辑器,下面是对vim编辑器的化:

```
[dmtsai@study ~]$ vim ~/.vimrc
"这个档案的双引号 (") 是註解
set hlsearch      "高亮度反白
set backspace=2   "可随时用倒退键删除
set autoindent    "自动缩排
set ruler         "可显示最后一列的状态
set showmode      "左下角那一列的状态
set nu            "可以在每一列的最前面显示行号啦!
set bg=dark       "显示不同的底色色调
syntax on         "进行语法检验, 颜色显示。
```

二、vi/vim编辑器常用功能:

1. 删除文本 在命令行模式中 使用 : 1,\$d 进行从第一行到最后的删除
2. vi编辑器中在命令行模式下输入G可以直接跳转到页面的底部, 在命令行模式下输入1G可以跳转到面的头部位置, 输入gg也可以回到第一行
- 3.命令: 『 :29,41s/man/MAN/gc 』把29-41中的man替换为MAN了
- 4.命令模式中: dG 也可以删除全文
- 5.搜索替换
| :s/one/two | 将当前光标所在行的第一个one替换成two |
| :s/one/two/g | 将当前光标所在行的所有one替换成two |
| :%s/one/two/g | 将全文中的所有one替换成two |

2.1 vim编辑器

命令 作用

- dd 删除(剪切)光标所在整行。
- 5dd 删除(剪切)从光标处开始的5行。
- yy 复制光标所在整行。
- 5yy 复制从光标处开始的5行。
- n 显示搜索命令定位到的下一个字符串。
- N 显示搜索命令定位到的上一个字符串。
- u 撤销上一步的操作
- p 将之前删除 (dd) 或复制 (yy) 过的数据粘贴到光标后。
- ?字符串 在文本中从下至上搜索该字符串。
- /字符串 在文本中从上至下搜索该字符串。

2.2 末行模式

末行模式用于保存或退出文件，能够设置vim编辑器的整体使用环境，让用户执行外部Linux命令或跳到特定的行数。末行模式切换的方式就是在命令模式中输入一个冒号就可以的，具体的命令如下表所示：

命令	作用
:w	保存
:q	退出
:q!	强制退出（放弃对文本的修改内容）
:wq!	强制保存退出
:set nu	显示行号
:set nonu	不显示行号
:命令	执行该命令
:整数	跳转到该行
:s/one/two	将当前光标所在行的第一个one替换成two
:s/one/two/g	将当前光标所在行的所有one替换成two
:%s/one/two/g	将全文中的所有one替换成two

2.3 移动光标的方法

h 或 向左方向键(←) 光标向左移动一个字元

j 或 向下方向键(↓) 光标向下移动一个字元

k 或 向上方向键(↑) 光标向上移动一个字元

l 或 向右方向键(→) 光标向右移动一个字元

如果你将右手放在键盘上的话，你会发现 hjkl 是排列在一起的，因此可以使用这四个按钮来移动光标。如果想要进行多次移动的话，例如向下移动 30 列，可以使用 "30j" 或 "30↓" 的组合按键，亦即加想要进行的次数(数字)后，按下动作即可！

[Ctrl] + [f] 萤幕『向下』移动一页，相当於 [Page Down]按键 (常用)

[Ctrl] + [b] 萤幕『向上』移动一页，相当於 [Page Up] 按键 (常用)

[Ctrl] + [d] 萤幕『向下』移动半页

[Ctrl] + [u] 萤幕『向上』移动半页

•

光标移动到非空白字元的下一列

•

光标移动到非空白字元的上一列

n + space 那个 n 表示『数字』，例如 20。按下数字后再按空白键，光标会向右移动这一列的 n 字元。例如 20 + space 则光标会向后面移动 20 个字元距离。

0 或功能键[Home] 这是数字『0』：移动到这一列的最前面字元处 (常用)

\$ 或功能键[End] 移动到这一列的最后面字元处(常用)

H 游标移动到萤幕的最上方那一列的第一个字元

M 游标移动到萤幕的中央那一列的第一个字元

L 游标移动到萤幕的最下方那一列的第一个字元

G 移动到档案的最后一列(常用)

nG n 為数字。移动到档案的第 n 列。例如 20G 则会移动到档案的第 20 列(可配合 :set nu)

gg 移动到档案的第一列, 相当於 1G 啊! (常用)

n + Enter n 為数字。游标向下移动 n 列(常用)

2.4 搜寻与取代

/word 向游标之下寻找一个名称為 word 的字串。例如要在档案内搜寻 vbird 这个字串, 就输入 /vbird 即可! (常用)

?word 向游标之上寻找一个字串名称為 word 的字串。

n 这个 n 是英文按键。代表『重复前一个搜寻的动作』。举例来说, 如果刚刚我们执行 /vbird 去搜寻 vbird 这个字串, 则按下 n 后, 会向下继续搜寻下一个名称為 vbird 的字串。如果是执行 ?vbird 的话, 那麼按下 n 则会向上继续搜寻名称為 vbird 的字串!

N 这个 N 是英文按键。与 n 刚好相反, 為『反向』进行前一个搜寻动作。例如 /vbird 后, 按下 N 则表示『向上』搜寻 vbird 。

使用 /word 配合 n 及 N 是非常有帮助的! 可以让你重复的找到一些你搜寻的关键字!

:n1,n2s/word1/word2/g n1 与 n2 為数字。在第 n1 与 n2 列之间寻找 word1 这个字串, 并将该串取代為 word2! 举例来说, 在 100 到 200 列之间搜寻 vbird 并取代為 VBIRD 则:

『:100,200s/vbird/VBIRD/g』。(常用)

:1,\$s/word1/word2/g 从第一列到最后一列寻找 word1 字串, 并将该字串取代為 word2! (常用)

:1,\$s/word1/word2/gc 从第一列到最后一列寻找 word1 字串, 并将该字串取代為 word2! 且在代前显示提示字元给使用者确认(confirm) 是否需要取代! (常用)

删除、複製与贴上

x, X 在一列字当中, x 為向后删除一个字元(相当於 [del] 按键), X 為向前删除一个字元(相当於 [backspace] 亦即是倒退键) (常用)

nx n 為数字, 连续向后删除 n 个字元。举例来说, 我要连续删除 10 个字元, 『10x』。

dd 删除游标所在的那一整列(常用)

ndd n 為数字。删除游标所在的向下 n 列, 例如 20dd 则是删除 20 列(常用)

d1G 删除游标所在到第一列的所有资料

dG 删除游标所在到最后一列的所有资料

d\$ 删除游标所在处, 到该列的最后一个字元

d0 那个是数字的 0, 删除游标所在处, 到该列的最前面一个字元

yy 複製游标所在的那一整列(常用)

nyy n 為数字。複製游标所在的向下 n 列, 例如 20yy 则是複製 20 列(常用)

y1G 複製游标所在到第一列的所有资料

yG 複製游标所在列到最后一列的所有资料

y0 複製游标所在的那个字元到该列行首的所有资料

y\$ 複製游标所在的那个字元到该列行尾的所有资料

p, P p 為将已複製的资料在游标下一列贴上, P 則為贴在游标上一列! 举例来说, 我目前游标在第 0 列, 且已经複製了 10 列资料。则按下 p 后, 那 10 列资料会贴在原本的 20 列之后, 亦即由 21 开始贴。但如果是按下 P 呢? 那麼原本的第 20 列会被推到变成 30 列。(常用)

J 将游标所在列与下一列的资料结合成同一列

c 重复删除多个资料, 例如向下删除 10 列, [10cj]

u 復原前一个动作。(常用)

[Ctrl]+r 重做上一个动作。(常用)

这个 u 与 [Ctrl]+r 是很常用的指令! 一个是復原, 另一个则是重做一次~ 利用这两个功能按键, 你编辑, 嘿嘿! 很快乐的啦!

. 不要怀疑! 这就是小数点! 意思是重复前一个动作的意思。如果你想要重复删除、重复贴上等等作, 按下小数点『.』就好了! (常用)

三、一般指令模式切换到编辑模式的可用的按钮说明

进入插入或取代的编辑模式

i, I 进入插入模式(Insert mode):

i 為『从目前游标所在处插入』, I 為『在目前所在列的第一个非空白字元处开始插入』。(常用)

a, A 进入插入模式(Insert mode):

a 為『从目前游标所在的下一个字元处开始插入』, A 為『从游标所在列的最后一个字元处开始插入』。(常用)

o, O 进入插入模式(Insert mode):

这是英文字母 o 的大小写。o 為『在目前游标所在的下一列处插入新的一列』; O 為在目前游标所处的上一列插入新的一列! (常用)

r, R 进入取代模式(Replace mode):

r 只会取代游标所在的那一个字元一次; R 会一直取代游标所在的文字, 直到按下 ESC 為止; (常用)

上面这些按键中, 在 vi 画面的左下角处会出现『--INSERT--』或『--REPLACE--』的字样。由名称知道该动作了吧!! 特别注意的是, 我们上面也提过了, 你想要在档案裡面输入字元时, 一定要在左下角处看到 INSERT 或 REPLACE 才能输入喔!

[Esc] 退出编辑模式, 回到一般指令模式中(常用)

四、一般指令模式切换到指令列模式的可用按钮说明

指令列模式的储存、离开等指令

:w 将编辑的资料写入硬碟档案中(常用)

:w! 若档案属性為『唯读』时, 强制写入该档案。不过, 到底能不能写入, 还是跟你对该档案的档权限有关啊!

:q 离开 vi (常用)

:q! 若曾修改过档案, 又不想储存, 使用! 為强制离开不储存档案。

注意一下啊，那个惊叹号 (!) 在 vi 当中，常常具有『强制』的意思～

:wq 储存后离开，若为 :wq! 则为强制储存后离开 (常用)

ZZ 这是大写的 Z 喔！若档案没有更动，则不储存离开，若档案已经被更动过，则储存后离开！

:w [filename] 将编辑的资料储存成另一个档案 (类似另存新档)

:r [filename] 在编辑的资料中，读入另一个档案的资料。亦即将『filename』这个档案内容加到标所在列后面

:n1,n2 w [filename] 将 n1 到 n2 的内容储存成 filename 这个档案。

:! command 暂时离开 vi 到指令列模式下执行 command 的显示结果！例如

『:! ls /home』即可在 vi 当中察看 /home 底下以 ls 输出的档案资讯！

vim 环境的变更

:set nu 显示行号，设定之后，会在每一列的字首显示该列的行号

:set nonu 与 set nu 相反，为取消行号！

五、同时打开多个文件

使用 vim 后面同时接好几个档案来同时开启喔！相关的按键有：多档案编辑的按键

:n 编辑下一个档案

:N 编辑上一个档案

:files 列出目前这个 vim 的开启的所有档案

六、多视窗情况下的按键功能

:sp [filename] 开启一个新视窗，如果有加 filename，表示在新视窗开启一个新档案，否则表示两视窗为同一个档案内容(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 (或向方向键)，则游标可移动到下方的视窗。

[ctrl]+w+k

[ctrl]+w+↑ 同上，不过游标移动到上面的视窗。

[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视窗，那么利用 [ctrl]+w+↓ 移动到下方视窗后，按下 :q 即可离开，也可以按下 [ctrl]+w+q 啊！

建议可以记忆的主要 vim 补齐功能，大致有底下几个：

组合按钮 补齐的内容

[ctrl]+x -> [ctrl]+n 透过目前正在编辑的这个『档案的内容文字』作为关键字，予以补齐

[ctrl]+x -> [ctrl]+f 以当前目录内的『档名』作为关键字，予以补齐

[ctrl]+x -> [ctrl]+o 以副档名作为语法补充，以 vim 内建的关键字，予以补齐