# Vim常用命令

## 操作命令

### 拷贝和粘贴

yy 拷贝当前行

nyy 拷贝当前后开始的n行，比如2yy拷贝当前行及其下一行。

p 在当前光标后粘贴,如果之前使用了yy命令来复制一行，那么就在当前行的下一行粘贴。

shift+p 在当前行前粘贴

:1,10 co 20 将1-10行插入到第20行之后。

:1,$ co $ 将整个文件复制一份并添加到文件尾部。

正常模式下按v（逐字）或V（逐行）进入可视模式，然后用jklh命令移动即可选择某些行或字符，再按y即可复制

ddp交换当前行和其下一行

xp交换当前字符和其后一个字符

# 文档修改

## vim修改文件的内容（改）

https://www.cnblogs.com/xkxk/p/5501690.html

命令：vim 文件

示例：编辑/test下的aaa.txt文件，使用vim aaa.txt

但此时并不能编辑，因为此时处于命令模式，点击键盘i/a/o进入编辑模式，可以 编辑文件

编辑完成后，按下Esc，退回命令模式

此时文件虽然已经编辑完成，但是没有保存，需输入冒号：进入底行模式，在底行模 式下输入wq代表写入内容并退出，即保存；输入q!代表强制退出不保存。

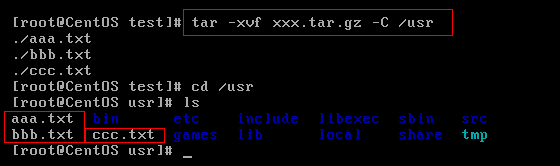
**总结：**

vim编辑器是Linux中的强大组件，是vi编辑器的加强版，vim编辑器的命令和快捷方式有很多，但此处不一一阐述，大家也无需研究的很透彻，使用vim编辑修改文件的方式基本会使用就可以了。附：[《vi使用方法介绍》](资料/04-vi使用方法详细介绍.docx)和[《vim命令合集》](资料/05-Vim命令合集.docx)

关于vim使用过程：

在实际开发中，使用vim编辑器主要作用就是修改配置文件

**vim 文件------>进入文件----->命令模式------>按i进入编辑模式----->编辑文件 ------->按Esc进入底行模式----->输入:wq/q!**



# vi使用方法详细介绍

vi编辑器是所有Unix及Linux系统下标准的编辑器，它的强大不逊色于任何最新的文本编辑器，这里只是简单地介绍一下它的用法和一小部分指令。由于对Unix及Linux系统的任何版本，vi编辑器是完全相同的，因此您可以在其他任何介绍vi的地方进一步了解它。Vi也是Linux中最基本的文本编辑器，学会它后，您将在Linux的世界里畅行无阻。   
  
1、vi的基本概念   
　　基本上vi可以分为三种状态，分别是命令模式（command mode）、插入模式（Insert mode）和底行模式（last line mode），各模式的功能区分如下：   
    1) 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。   
    2) 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。   
    3) 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。   
  
    不过一般我们在使用时把vi简化成两个模式，就是将底行模式（last line mode）也算入命令行模式command mode）。   
2、vi的基本操作   
a) 进入vi   
   　在系统提示符号输入vi及文件名称后，就进入vi全屏幕编辑画面：  
　　　$ vi myfile   
　　不过有一点要特别注意，就是您进入vi之后，是处于「命令行模式（command mode）」，您要切换到「插入模式（Insert mode）」才能够输入文字。初次使用vi的人都会想先用上下左右键移动光标，结果电脑一直哔哔叫，把自己气个半死，所以进入vi后，先不要乱动，转换到「插入模式（Insert mode）」再说吧！   
  
b) 切换至插入模式（Insert mode）编辑文件   
　　在「命令行模式（command mode）」下按一下字母「i」就可以进入「插入模式（Insert mode）」，这时候你就可以开始输入文字了。   
  
c) Insert 的切换   
　　您目前处于「插入模式（Insert mode）」，您就只能一直输入文字，如果您发现输错了字！想用光标键往回移动，将该字删除，就要先按一下「ESC」键转到「命令行模式（command mode）」再删除文字。   
  
d) 退出vi及保存文件   
　　在「命令行模式（command mode）」下，按一下「：」冒号键进入「Last line mode」，例如：   
: w filename （输入 「w filename」将文章以指定的文件名filename保存）   
: wq (输入「wq」，存盘并退出vi)   
: q! (输入q!， 不存盘强制退出vi)   
  
3、命令行模式（command mode）功能键   
1）. 插入模式   
       按「i」切换进入插入模式「insert mode」，按"i"进入插入模式后是从光标当前位置开始输入文件；   
　　按「a」进入插入模式后，是从目前光标所在位置的下一个位置开始输入文字；   
　　按「o」进入插入模式后，是插入新的一行，从行首开始输入文字。   
  
2）. 从插入模式切换为命令行模式   
      按「ESC」键。   
  
3）. 移动光标   
　　vi可以直接用键盘上的光标来上下左右移动，但正规的vi是用小写英文字母「h」、「j」、「k」、「l」，分别控制光标左、下、上、右移一格。   
　　按「ctrl」+「b」：屏幕往"后"移动一页。   
　　按「ctrl」+「f」：屏幕往"前"移动一页。   
　　按「ctrl」+「u」：屏幕往"后"移动半页。   
　　按「ctrl」+「d」：屏幕往"前"移动半页。   
　　按数字「0」：移到文章的开头。   
　　按「G」：移动到文章的最后。   
　　按「$」：移动到光标所在行的"行尾"。   
　　按「^」：移动到光标所在行的"行首"   
　　按「w」：光标跳到下个字的开头   
　　按「e」：光标跳到下个字的字尾   
　　按「b」：光标回到上个字的开头   
　　按「#l」：光标移到该行的第#个位置，如：5l,56l。   
  
4）. 删除文字   
　　「x」：每按一次，删除光标所在位置的"后面"一个字符。   
　　「#x」：例如，「6x」表示删除光标所在位置的"后面"6个字符。   
　　「X」：大写的X，每按一次，删除光标所在位置的"前面"一个字符。   
　　「#X」：例如，「20X」表示删除光标所在位置的"前面"20个字符。   
　　「dd」：删除光标所在行。   
　　「#dd」：从光标所在行开始删除#行   
  
5）. 复制   
　　「yw」：将光标所在之处到字尾的字符复制到缓冲区中。   
　　「#yw」：复制#个字到缓冲区   
　　「yy」：复制光标所在行到缓冲区。   
　　「#yy」：例如，「6yy」表示拷贝从光标所在的该行"往下数"6行文字。   
　　「p」：将缓冲区内的字符贴到光标所在位置。注意：所有与"y"有关的复制命令都必须与"p"配合才能完成复制与粘贴功能。   
  
6）. 替换   
　　「r」：替换光标所在处的字符。   
　　「R」：替换光标所到之处的字符，直到按下「ESC」键为止。   
  
7）. 回复上一次操作   
　　「u」：如果您误执行一个命令，可以马上按下「u」，回到上一个操作。按多次"u"可以执行多次回复。   
  
8）. 更改   
　　「cw」：更改光标所在处的字到字尾处   
　　「c#w」：例如，「c3w」表示更改3个字   
  
9）. 跳至指定的行   
　　「ctrl」+「g」列出光标所在行的行号。   
　　「#G」：例如，「15G」，表示移动光标至文章的第15行行首。   
4、Last line mode下命令简介   
　　在使用「last line mode」之前，请记住先按「ESC」键确定您已经处于「command mode」下后，再按「：」冒号即可进入「last line mode」。   
  
A) 列出行号   
  
　「set nu」：输入「set nu」后，会在文件中的每一行前面列出行号。   
  
B) 跳到文件中的某一行   
  
　「#」：「#」号表示一个数字，在冒号后输入一个数字，再按回车键就会跳到该行了，如输入数字15，再回车，就会跳到文章的第15行。   
  
C) 查找字符   
  
　「/关键字」：先按「/」键，再输入您想寻找的字符，如果第一次找的关键字不是您想要的，可以一直按「n」会往后寻找到您要的关键字为止。   
  
　「?关键字」：先按「?」键，再输入您想寻找的字符，如果第一次找的关键字不是您想要的，可以一直按「n」会往前寻找到您要的关键字为止。   
  
D) 保存文件   
  
　「w」：在冒号输入字母「w」就可以将文件保存起来。   
  
E) 离开vi   
  
　「q」：按「q」就是退出，如果无法离开vi，可以在「q」后跟一个「!」强制离开vi。   
  
　「qw」：一般建议离开时，搭配「w」一起使用，这样在退出的时候还可以保存文件。   
  
5、vi命令列表   
1、下表列出命令模式下的一些键的功能：   
  
h   
左移光标一个字符   
  
l   
右移光标一个字符   
  
k   
光标上移一行   
  
j   
光标下移一行   
  
^   
光标移动至行首   
  
0   
数字"0"，光标移至文章的开头   
  
G   
光标移至文章的最后   
  
$   
光标移动至行尾   
  
Ctrl+f   
向前翻屏   
  
Ctrl+b   
向后翻屏   
  
Ctrl+d   
向前翻半屏   
  
Ctrl+u   
向后翻半屏   
  
i   
在光标位置前插入字符   
  
a   
在光标所在位置的后一个字符开始增加   
  
o   
插入新的一行，从行首开始输入   
  
ESC   
从输入状态退至命令状态   
  
x   
删除光标后面的字符   
  
#x   
删除光标后的＃个字符   
  
X   
(大写X)，删除光标前面的字符   
  
#X   
删除光标前面的#个字符   
  
dd   
删除光标所在的行   
  
#dd   
删除从光标所在行数的#行   
  
yw   
复制光标所在位置的一个字   
  
#yw   
复制光标所在位置的#个字   
  
yy   
复制光标所在位置的一行   
  
#yy   
复制从光标所在行数的#行   
  
p   
粘贴   
  
u   
取消操作   
  
cw   
更改光标所在位置的一个字   
  
#cw   
更改光标所在位置的#个字   
  
  
2、下表列出行命令模式下的一些指令   
w filename   
储存正在编辑的文件为filename   
  
wq filename   
储存正在编辑的文件为filename，并退出vi   
  
q!   
放弃所有修改，退出vi   
  
set nu   
显示行号   
  
/或?   
查找，在/后输入要查找的内容   
  
n   
与/或?一起使用，如果查找的内容不是想要找的关键字，按n或向后（与/联用）或向前（与?联用）继续查找，直到找到为止。   
  
  
对于第一次用vi，有几点注意要提醒一下：   
1、用vi打开文件后，是处于「命令行模式（command mode）」，您要切换到「插入模式（Insert mode）」才能够输入文字。切换方法：在「命令行模式（command mode）」下按一下字母「i」就可以进入「插入模式（Insert mode）」，这时候你就可以开始输入文字了。   
2、编辑好后，需从插入模式切换为命令行模式才能对文件进行保存，切换方法：按「ESC」键。   
3、保存并退出文件：在命令模式下输入:wq即可！（别忘了wq前面的）

**vi使用手册**

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | |  | | --- | | VI是unix上最常用的文本编辑工具，作为unix软件测试人员，有必要熟练掌握它。  进入vi的命令 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 +/pattern filename：打开文件，并将光标置于第一个与pattern匹配的串处 vi -r filename ：在上次正用vi编辑时发生系统崩溃，恢复filename vi filename....filename ：打开多个文件，依次编辑  移动光标类命令 h ：光标左移一个字符 l ：光标右移一个字符 space：光标右移一个字符 Backspace：光标左移一个字符 k或Ctrl+p：光标上移一行 j或Ctrl+n ：光标下移一行 Enter ：光标下移一行 w或W ：光标右移一个字至字首 b或B ：光标左移一个字至字首 e或E ：光标右移一个字j至字尾 ) ：光标移至句尾 ( ：光标移至句首 }：光标移至段落开头 {：光标移至段落结尾 nG：光标移至第n行首 n+：光标下移n行 n-：光标上移n行 n$：光标移至第n行尾 H ：光标移至屏幕顶行 M ：光标移至屏幕中间行 L ：光标移至屏幕最后行 0：（注意是数字零）光标移至当前行首 $：光标移至当前行尾  屏幕翻滚类命令 Ctrl+u：向文件首翻半屏 Ctrl+d：向文件尾翻半屏 Ctrl+f：向文件尾翻一屏 Ctrl＋b；向文件首翻一屏 nz：将第n行滚至屏幕顶部，不指定n时将当前行滚至屏幕顶部。  插入文本类命令 i ：在光标前 I ：在当前行首 a：光标后 A：在当前行尾 o：在当前行之下新开一行 O：在当前行之上新开一行 r：替换当前字符 R：替换当前字符及其后的字符，直至按ESC键 s：从当前光标位置处开始，以输入的文本替代指定数目的字符 S：删除指定数目的行，并以所输入文本代替之 ncw或nCW：修改指定数目的字 nCC：修改指定数目的行  删除命令 ndw或ndW：删除光标处开始及其后的n-1个字 do：删至行首 d$：删至行尾 ndd：删除当前行及其后n-1行 x或X：删除一个字符，x删除光标后的，而X删除光标前的 Ctrl+u：删除输入方式下所输入的文本  搜索及替换命令 : /pattern：从光标开始处向文件尾搜索pattern ?pattern：从光标开始处向文件首搜索pattern n：在同一方向重复上一次搜索命令 N：在反方向上重复上一次搜索命令 ：s/p1/p2/g：将当前行中所有p1均用p2替代 ：n1,n2s/p1/p2/g：将第n1至n2行中所有p1均用p2替代 ：g/p1/s//p2/g：将文件中所有p1均用p2替换  选项设置 all：列出所有选项设置情况 term：设置终端类型 ignorance：在搜索中忽略大小写 list：显示制表位(Ctrl+I)和行尾标志（$) number：显示行号 report：显示由面向行的命令修改过的数目 terse：显示简短的警告信息 warn：在转到别的文件时若没保存当前文件则显示NO write信息 nomagic：允许在搜索模式中，使用前面不带“\”的特殊字符 nowrapscan：禁止vi在搜索到达文件两端时，又从另一端开始 mesg：允许vi显示其他用户用write写到自己终端上的信息  最后行方式命令 ：n1,n2 co n3：将n1行到n2行之间的内容拷贝到第n3行下 ：n1,n2 m n3：将n1行到n2行之间的内容移至到第n3行下 ：n1,n2 d ：将n1行到n2行之间的内容删除 ：w ：保存当前文件 ：e filename：打开文件filename进行编辑 ：x：保存当前文件并退出 ：q：退出vi ：q!：不保存文件并退出vi ：!command：执行shell命令command ：n1,n2 w!command：将文件中n1行至n2行的内容作为command的输入并执行之，若不指 定n1，n2，则表示将整个文件内容作为command的输入 ：r!command：将命令command的输出结果放到当前行 。 | | |

# Vim

[Vim命令合集](http://www.cnblogs.com/softwaretesting/archive/2011/07/12/2104435.html)

**命令历史**

以:和/开头的命令都有历史纪录，可以首先键入:或/然后按上下箭头来选择某个历史命令。

**启动vim**

在命令行窗口中输入以下命令即可

vim 直接启动vim

vim filename 打开vim并创建名为filename的文件

**文件命令**

打开单个文件

vim file

同时打开多个文件

vim file1 file2 file3 ...

在vim窗口中打开一个新文件

:open file

在新窗口中打开文件

:split file

切换到下一个文件

:bn

切换到上一个文件

:bp

查看当前打开的文件列表，当前正在编辑的文件会用[]括起来。

:args

打开远程文件，比如ftp或者share folder

:e ftp://192.168.10.76/abc.txt

:e \\qadrive\test\1.txt

**vim的模式**

正常模式（按Esc或Ctrl+[进入） 左下角显示文件名或为空

插入模式（按i键进入） 左下角显示--INSERT--

可视模式（不知道如何进入） 左下角显示--VISUAL--

**导航命令**

% 括号匹配

**插入命令**

i 在当前位置生前插入

I 在当前行首插入

a 在当前位置后插入

A 在当前行尾插入

o 在当前行之后插入一行

O 在当前行之前插入一行

**查找命令**

/text　　查找text，按n健查找下一个，按N健查找前一个。

?text　　查找text，反向查找，按n健查找下一个，按N健查找前一个。

vim中有一些特殊字符在查找时需要转义　　.\*[]^%/?~$

:set ignorecase　　忽略大小写的查找

:set noignorecase　　不忽略大小写的查找

查找很长的词，如果一个词很长，键入麻烦，可以将光标移动到该词上，按\*或#键即可以该单词进行搜索，相当于/搜索。而#命令相当于?搜索。

:set hlsearch　　高亮搜索结果，所有结果都高亮显示，而不是只显示一个匹配。

:set nohlsearch　　关闭高亮搜索显示

:nohlsearch　　关闭当前的高亮显示，如果再次搜索或者按下n或N键，则会再次高亮。

:set incsearch　　逐步搜索模式，对当前键入的字符进行搜索而不必等待键入完成。

:set wrapscan　　重新搜索，在搜索到文件头或尾时，返回继续搜索，默认开启。

**替换命令**

ra 将当前字符替换为a，当期字符即光标所在字符。

s/old/new/ 用old替换new，替换当前行的第一个匹配

s/old/new/g 用old替换new，替换当前行的所有匹配

%s/old/new/ 用old替换new，替换所有行的第一个匹配

%s/old/new/g 用old替换new，替换整个文件的所有匹配

:10,20 s/^/    /g 在第10行知第20行每行前面加四个空格，用于缩进。

ddp 交换光标所在行和其下紧邻的一行。

**移动命令**

h 左移一个字符

l 右移一个字符，这个命令很少用，一般用w代替。

k 上移一个字符

j 下移一个字符

以上四个命令可以配合数字使用，比如20j就是向下移动20行，5h就是向左移动5个字符，在Vim中，很多命令都可以配合数字使用，比如删除10个字符10x，在当前位置后插入3个！，3a！<Esc>，这里的Esc是必须的，否则命令不生效。

w 向前移动一个单词（光标停在单词首部），如果已到行尾，则转至下一行行首。此命令快，可以代替l命令。

b 向后移动一个单词 2b 向后移动2个单词

e，同w，只不过是光标停在单词尾部

ge，同b，光标停在单词尾部。

^ 移动到本行第一个非空白字符上。

0（数字0）移动到本行第一个字符上，

<HOME> 移动到本行第一个字符。同0健。

$ 移动到行尾 3$ 移动到下面3行的行尾

gg 移动到文件头。 = [[

G（shift + g） 移动到文件尾。 = ]]

f（find）命令也可以用于移动，fx将找到光标后第一个为x的字符，3fd将找到第三个为d的字符。

F 同f，反向查找。

跳到指定行，冒号+行号，回车，比如跳到240行就是 :240回车。另一个方法是行号+G，比如230G跳到230行。

Ctrl + e 向下滚动一行

Ctrl + y 向上滚动一行

Ctrl + d 向下滚动半屏

Ctrl + u 向上滚动半屏

Ctrl + f 向下滚动一屏

Ctrl + b 向上滚动一屏

**撤销和重做**

u 撤销（Undo）

U 撤销对整行的操作

Ctrl + r 重做（Redo），即撤销的撤销。

**删除命令**

x 删除当前字符

3x 删除当前光标开始向后三个字符

X 删除当前字符的前一个字符。X=dh

dl 删除当前字符， dl=x

dh 删除前一个字符

dd 删除当前行

dj 删除上一行

dk 删除下一行

10d 删除当前行开始的10行。

D 删除当前字符至行尾。D=d$

d$ 删除当前字符之后的所有字符（本行）

kdgg 删除当前行之前所有行（不包括当前行）

jdG（jd shift + g）   删除当前行之后所有行（不包括当前行）

:1,10d 删除1-10行

:11,$d 删除11行及以后所有的行

:1,$d 删除所有行

J(shift + j)　　删除两行之间的空行，实际上是合并两行。

**剪切命令**

正常模式下按v（逐字）或V（逐行）进入可视模式，然后用jklh命令移动即可选择某些行或字符，再按d即可剪切

ndd 剪切当前行之后的n行。利用p命令可以对剪切的内容进行粘贴

:1,10d 将1-10行剪切。利用p命令可将剪切后的内容进行粘贴。

:1, 10 m 20 将第1-10行移动到第20行之后。

**退出命令**

:wq 保存并退出

ZZ 保存并退出

:q! 强制退出并忽略所有更改

:e! 放弃所有修改，并打开原来文件。

**窗口命令**

:split或new 打开一个新窗口，光标停在顶层的窗口上

:split file或:new file 用新窗口打开文件

split打开的窗口都是横向的，使用vsplit可以纵向打开窗口。

Ctrl+ww 移动到下一个窗口

Ctrl+wj 移动到下方的窗口

Ctrl+wk 移动到上方的窗口

关闭窗口

:close 最后一个窗口不能使用此命令，可以防止意外退出vim。

:q 如果是最后一个被关闭的窗口，那么将退出vim。

ZZ 保存并退出。

关闭所有窗口，只保留当前窗口

:only

录制宏

按q键加任意字母开始录制，再按q键结束录制（这意味着vim中的宏不可嵌套），使用的时候@加宏名，比如qa。。。q录制名为a的宏，@a使用这个宏。

**执行shell命令**

:!command

:!ls 列出当前目录下文件

:!perl -c script.pl 检查perl脚本语法，可以不用退出vim，非常方便。

:!perl script.pl 执行perl脚本，可以不用退出vim，非常方便。

:suspend或Ctrl - Z 挂起vim，回到shell，按fg可以返回vim。

**注释命令**

perl程序中#开始的行为注释，所以要注释某些行，只需在行首加入#

3,5 s/^/#/g 注释第3-5行

3,5 s/^#//g 解除3-5行的注释

1,$ s/^/#/g 注释整个文档。

:%s/^/#/g 注释整个文档，此法更快。

**帮助命令**

:help or F1 显示整个帮助

:help xxx 显示xxx的帮助，比如 :help i, :help CTRL-[（即Ctrl+[的帮助）。

:help 'number' Vim选项的帮助用单引号括起

:help <Esc> 特殊键的帮助用<>扩起

:help -t Vim启动参数的帮助用-

：help i\_<Esc> 插入模式下Esc的帮助，某个模式下的帮助用模式\_主题的模式

帮助文件中位于||之间的内容是超链接，可以用Ctrl+]进入链接，Ctrl+o（Ctrl + t）返回

**其他非编辑命令**

. 重复前一次命令

:set ruler?　　查看是否设置了ruler，在.vimrc中，使用set命令设制的选项都可以通过这个命令查看

:scriptnames　　查看vim脚本文件的位置，比如.vimrc文件，语法文件及plugin等。

:set list 显示非打印字符，如tab，空格，行尾等。如果tab无法显示，请确定用set lcs=tab:>-命令设置了.vimrc文件，并确保你的文件中的确有tab，如果开启了expendtab，那么tab将被扩展为空格。

Vim教程

在Unix系统上

$ vimtutor

在Windows系统上

:help tutor

:syntax 列出已经定义的语法项

:syntax clear 清除已定义的语法规则

:syntax case match 大小写敏感，int和Int将视为不同的语法元素

:syntax case ignore 大小写无关，int和Int将视为相同的语法元素，并使用同样的配色方案