**Linux的vi**编辑器****

# vi及vim命令介绍

**vi命令**是UNIX操作系统和类UNIX操作系统中最通用的全屏幕纯文本编辑器。Linux中的vi编辑器叫**vim**，它是vi的增强版（vi Improved），与vi编辑器完全兼容，而且实现了很多增强功能。

# vi编辑器的编辑模式与命令模式

## 编辑模式与命令模式

**vi编辑器**支持**编辑模式**和**命令模式**，编辑模式下可以完成文本的编辑功能，命令模式下可以完成对文件的操作命令，要正确使用vi编辑器就必须熟练掌握着两种模式的切换。

## 编辑模式与命令模式的切换

默认情况下，打开vi编辑器后自动进入命令模式。

从编辑模式切换到命令模式使用**“Esc”键**(Escape)，从命令模式切换到编辑模式使用**“A”、“a”、“O”、“o”、“I”、“i”“s”“S”**键。(大小写a、i、o、s)

在命令模式下进入编辑模式，有6个命令，对应的编辑位置不同：

分别对应当前字符前后、当前行首与行尾、当前行前与行后插入一行；

a：在**当前字符后**添加文本；

A：在**行末**添加文本；

i：在**当前字符前**插入文本；

I：在**行首**插入文本；

o：在**当前行后面**插入一空行，添加文本；

O：在**当前行前面**插入一空行，添加文本；

s：删除当前字符添加文本；

S：删除当前行添加文本；

# vi编辑器的内置命令

vi编辑器提供了丰富的内置命令，有些内置命令使用**键盘组合键**即可完成，有些内置命令则需要以**冒号“：”**开头输入。具体来介绍常用的内置命令。

# 进入与退出vi编辑器命令

## 进入vi编辑器：

**vi [选项] [参数]或vim [选项] [参数]**

**+<行号>：从指定行号的行开始先是文本内容**；

-b：以**二进制模式**打开文件，用于编辑二进制文件和可执行文件；

-c<指令>：在完成对第一个文件编辑任务后，执行给出的指令；

-d：以**diff模式**打开文件，当多个文件编辑时，显示文件差异部分；

-l：使用lisp模式，打开“lisp”和“showmatch”；

-m：取消写文件功能，重设“write”选项；

-M：关闭修改功能；

-n：不实用缓存功能；

**-o<文件数目>**：指定同时打开指定数目的文件；

-R：以**只读方式**打开文件；

-s：安静模式，不显示指令的任何错误信息。

参数：就是文件列表，指定要编辑的文件列表。多个文件之间使用**空格**分隔开。

## 退出vi编辑器：

w表示**存盘(write)**，q表示**退出(quit)**，!表示**强制**。

**保存文件并退出**：**ZZ**、**:wq**、**:x**

**保存文件(存盘)不退出：:w**

**文件另存为不存盘不退出: :w file （注意：当前文件不存盘）**

**强制存盘操作：**:w！(如vi –R file 只读模式下，仍可以保存)

**正常退出：:q**

**强制退出(不保存)：:q!**

**放弃所有修改至上次存盘重新编辑：:e!**

# 光标的移动控制

在命令模式下，利用命令控制光标的移动。

方向键上下左右。

**左右移动**：**h**向右一个字符；**l或空格**向左一个字符；

**上下行移动：j**向下一行；**k**向上一行；

当前屏幕的位置：开头H、结尾L、中间M。

# 翻页：半页翻和全页翻

**Ctrl+U：向文件首翻半屏；<up>**

**Ctrl+D：向文件尾翻半屏；<down>**

**Ctrl+F：向文件尾翻一屏；**

**Ctrl+B：向文件首翻一屏；**

# 行号显示及跳转

### 最左端行号显示

**:set number**：在命令模式下，用于在最左端显示行号；

**:set nonumber**：在命令模式下，用于在最左端不显示行号；

### 跳转指定行行首

**:lineNum**：光标跳转到指定行的行首；

* 如果lineNum是正数，则调到对应行；(0和1一样都是首行);若数超出文本最大行数，则调到最后一行的行首，作用同:$
* 如果为负数，则调到当前行的前**|lineNum|**行。

**:$** ：光标跳转到最后一行的行首。

# 删除字符、单词、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；

**nx或nX**：删除n个字符；

dw：删除下一个单词；

ndw：删除光标后n个单词；

**常用的删除行**：

**dd**：删除当前行；

**ndd**：删除当前行及其后n-1行；(删除多行)如3dd

**D**：删除从当前光标到光标**所在行尾的全部字符**；

**dG** ：删除当前光标以后的所有内容；

db 删除光标前面的字  
ndb 从当前行开始往前删除n字  
:n,md 从第m行开始往前删除n行  
d或d$ 从光标处删除到行尾  
dcursor\_command 删除至光标命令处，如dG将从当前行删除至文件的末尾  
^h或backspace 插入时，删除前面的字符  
^w 插入时，删除前面的字

# 复制

**yy**：复制当前行；

nyy：将当前行及其下n行的内容保存到寄存器？中，其中？为一个字母，n为一个数字；

# 粘贴

**p**：粘贴文本操作，用于将缓存区的内容粘贴到当前光标所在**位置的下方**；

**P**：粘贴文本操作，用于将缓存区的内容粘贴到当前光标所在**位置的上方**；

# 查找与替换

## 查找字符串

### 全文查找

**/text 在文件中向前查找text  
?text 在文件中向后查找text**

**/字符串**：文本查找操作，用于从**当前光标所在位置**开始**向文件尾部**查找指定字符串的内容，查找的字符串会被加亮显示；

**?字符串**：文本查找操作，用于**从当前光标所在位置**开始**向文件头部查找**指定字符串的内容，查找的字符串会被加亮显示；

* n向同方向查找下一个；
* N向反方向查找下一个；

**注意**：**使用/str时，结合n和N就可以完成上下搜索;**使用N，自动转成?str;同样使用?str,再用n也会自动转成/str。

### 行查找： ftext 在当前行向前查找text Ftext 在当前行向后查找text ttext 在当前行向前查找text，并将光标定位在text的第一个字符 Ttext 在当前行向后查找text，并将光标定位在text的第一个字符

### 设置大小写敏感 :set ic 查找时忽略大小写 :set noic 查找时对大小写敏感

## 替换命令 :s/oldtext/newtext 在当前行,用newtext替换oldtext；

& 重复最后的:s命令;

以上两个命令结合使用。

:m,ns/oldtext/newtext 在m和n行之间，用newtext替换oldtext  
  
:g/text1/s/text2/text3 查找包含text1的行，用text3替换text2  
:g/text/command 在所有**包含text**的行运行command所表示的命令  
:v/text/command 在所有**不包含text**的行运行command所表示的命令

# 撤消与重复

**u 撤消最后一次修改  
U 撤消当前行的所有修改**  
**. 重复最后一次修改**, 以相反的方向重复前面的f、F、t或T查找命令  
; 重复前面的f、F、t或T查找命令  
"np 取回最后第n次的删除(缓冲区中存有一定次数的删除内容，一般为9)  
n 重复前面的**/或?**查找命令  
N 以相反方向重复前面的**/或?**命令

# 连接行

J 将下一行连接到当前行的末尾；

nJ 自动连接后面n行

# 显示文件名、总行数、当前行号等信息

**:f** ：在命令模式下，用于**显示当前的文件名、光标所在行的行号以及显示比例**；

# 将其他文件内容插入到本文件

**:r file** 读入文件file内容，并插在当前行后；

**:nr file** 读入文件file内容，并插在第n行后

**^v char** 插入时忽略char的指定意义，这是为了插入特殊字符

**结合:f 查看行号信息。**

# 打开并编辑指定名称的文件

:e文件名：在命令模式下，打开并编辑指定名称的文件；

:n：在命令模式下，如果同时打开多个文件，则继续编辑下一个文件；

# 在vi中删除文本 命令 删除操作修改vi文本 每个命令前面的数字表示该命令重复的次数 命令 替换操作 rchar 用char替换当前字符 R text escape 用text替换当前字符直到换下Esc键 stext escape 用text代替当前字符 S或cctext escape 用text代替整行 cwtext escape 将当前字改为text Ctext escape 将当前行余下的改为text cG escape 修改至文件的末尾 ccursor\_cmd text escape 从当前位置处到光标命令位置处都改为text

# 在vi中复制文本 命令 复制操作 yy 将当前行的内容放入临时缓冲区 nyy 将n行的内容放入临时缓冲区 p 将临时缓冲区中的文本放入光标后 P 将临时缓冲区中的文本放入光标前 dsfsd "(a-z)nyy 复制n行放入名字为圆括号内的可命名缓冲区，省略n表示当前行 "(a-z)ndd 删除n行放入名字为圆括号内的可命名缓冲区，省略n表示当前行 "(a-z)p 将名字为圆括号的可命名缓冲区的内容放入当前行后 "(a-z)P 将名字为圆括号的可命名缓冲区的内容放入当前行前

# vi中的选项 选项 作用 :set all 打印所有选项 :set nooption 关闭option选项 :set nu 每行前打印行号 :set showmode 显示是输入模式还是替换模式 :set noic 查找时忽略大小写 :set list 显示制表符(^I)和行尾符号 :set ts=8 为文本输入设置tab stops :set window=n 设置文本窗口显示n行

# vi的状态 选项 作用 :.= 打印当前行的行号 := 打印文件中的行数 ^g 显示文件名、当前的行号、文件的总行数和文件位置的百分比 :l 使用字母"l"来显示许多的[特殊字符](https://www.baidu.com/s?wd=%E7%89%B9%E6%AE%8A%E5%AD%97%E7%AC%A6&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YYrymLP1DkrjwbryDdnjKW0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjbzrH63PHfz)，如制表符和换行符

# 在文本中定位段落和放置标记 选项 作用 { 在第一列插入{来定义一个段落 [[ 回到段落的开头处 ]] 向前移到下一个段落的开头处 m(a-z) 用一个字母来标记当前位置，如用mz表示标记z '(a-z) 将光标移动到指定的标记，如用'z表示移动到z

# 光标放置与屏幕调整 选项 作用 H 将光标移动到屏幕的顶行 nH 将光标移动到屏幕顶行下的第n行 M 将光标移动到屏幕的中间 L 将光标移动到屏幕的底行 nL 将光标移动到屏幕底行上的第n行 ^e(ctrl+e) 将屏幕上滚一行 ^y 将屏幕下滚一行 ^u 将屏幕上滚半页 ^d 将屏幕下滚半页 ^b 将屏幕上滚一页 ^f 将屏幕下滚一页 ^l 重绘屏幕 z-return 将当前行置为屏幕的顶行 nz-return 将当前行下的第n行置为屏幕的顶行 z. 将当前行置为屏幕的中央 nz. 将当前行上的第n行置为屏幕的中央 z- 将当前行置为屏幕的底行 nz- 将当前行上的第n行置为屏幕的底行

# vi中的shell转义命令 选项 作用 :!command 执行shell的command命令，如:!ls :!! 执行前一个shell命令 :r!command 读取command命令的输入并插入，如:r!ls会先执行ls，然后读入内容 :w!command 将当前已编辑文件作为command命令的标准输入并执行command命令，如:w!grep all :cd directory 将当前工作目录更改为directory所表示的目录 :sh 将启动一个子shell，使用^d(ctrl+d)返回vi :so file 在[shell程序](https://www.baidu.com/s?wd=shell%E7%A8%8B%E5%BA%8F&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YYrymLP1DkrjwbryDdnjKW0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjbzrH63PHfz)file中读入和执行命令

# vi中的宏与缩写 (避免使用控制键和符号，不要使用字符K、V、g、q、v、\*、=和功能键) 选项 作用 :map key command\_seq 定义一个键来运行command\_seq，如:map e ea，无论什么时候都可以e移到一个字的末尾来追加文本 :map 在状态行显示所有已定义的宏 :umap key 删除该键的宏 :ab string1 string2 定义一个缩写，使得当插入string1时，用string2替换string1。当要插入文本时，键入string1然后按Esc键，系统就插入了string2 :ab 显示所有缩写 :una string 取消string的缩写

# 在vi中缩进文本 选项 作用 ^i(ctrl+i)或tab 插入文本时，插入移动的宽度，移动宽度是事先定义好的 :set ai 打开自动缩进 :set sw=n 将移动宽度设置为n个字符 n<< 使n行都向左移动一个宽度 n>> 使n行都向右移动一个宽度，例如3>>就将接下来的三行每行都向右移动一个移动宽度