

**vi** (vim) 是上Linux非常常用的编辑器，很多Linux发行版都默认安装了vi(vim)。vi(vim)命令繁多但是如果使用灵活之后将会大大提高效率。vi是"visual interface"的缩写，vim是vi IMproved(Improved)

vi有3个模式：**插入模式、命令模式、低行模式**

- 1) 插入模式：在此模式下可以输入字符，按ESC将回到命令模式。
- 2) 命令模式：可以移动光标、删除字符等。
- 3) 低行模式：可以保存文件、退出vi、设置vi、查找等功能(低行模式也可以看作是命令模式里的)。

**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 -o/O filename1 filename2 ... ：打开多个文件，依次进行编辑

**vi 关闭文件**

:w //保存文件  
:w vpser.net //保存至vpser.net文件  
:q //退出编辑器，如果文件已修改请使用下面的命令  
:q! //退出编辑器，且不保存  
:wq //退出编辑器，且保存文件

**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  
O：（注意是数字零）光标移至当前行首  
\$:光标移至当前行尾  
n+ //向下跳n行  
n- //向上跳n行  
nG //跳到行号为n的行  
G //跳至文件的底部

**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

复制、粘贴

- yy //将当前行复制到缓存区，也可以用 "ayy 复制，"a 为缓冲区，a也可以替换为a到z的任意字母，可以完成多个复制任务。
- nyy //将当前行向下n行复制到缓冲区，也可以用 "anyy 复制，"a 为缓冲区，a也可以替换为a到z的任意字母，可以完成多个复制任务。
- yw //复制从光标开始到词尾的字符。
- nyw //复制从光标开始的n个单词。
- y^ //复制从光标到行首的内容。
- y\$ //复制从光标到行尾的内容。
- p //粘贴剪切板里的内容在光标后，如果使用了前面的自定义缓冲区，建议使用"ap 进行粘贴。
- P //粘贴剪切板里的内容在光标前，如果使用了前面的自定义缓冲区，建议使用"aP 进行粘贴。

文本替换

- :s/old/new //用new替换行中首次出现的old
- :s/old/new/g //用new替换行中所有的old
- :n,m s/old/new/g //用new替换从n到m行里所有的old
- :%s/old/new/g //用new替换当前文件里所有的old

简单替换表达式

`:%s/four/4/g`

"%" 范围前缀表示在所有行中执行替换，最后的 "g" 标记表示替换行中的所有匹配点，如果仅仅对当前行进行操作，那么只要去掉%即可

如果你有一个像 "thirtyfour" 这样的单词，上面的命令会出错。这种情况下，这个单词会被替换成"thirty4"。要解决这个问题，用 "<"来指定匹配单词开头：

`:%s/<four/4/g`

显然，这样在处理 "fourty" 的时候还是会出错。用 ">" 来解决这个问题：

`:%s/<four>/4/g`

如果你在编码，你可能只想替换**注释**中的 "four"，而保留代码中的。由于这很难指定，可以在替换命令中加一个 "c" 标记，这样，Vim 会在每次替换前提示你：

`:%s/<four>/4/gc`

单词精确匹配替换

`sed -e "s/\<old\>/new/g" file`

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
- x //删除当前字符
- nx //删除从光标开始的n个字符
- dd //删除当前行
- ndd //向下删除当前行在内的n行
- u //撤销上一步操作
- U //撤销对当前行的所有操作

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

寄存器操作

- "?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
- "?nyw：将当前行及其下n个字符保存到寄存器？中，其中?为一个字母，n为一个数字
- "?nyl：将当前行及其下n个字符保存到寄存器？中，其中?为一个字母，n为一个数字
- "?p：取出寄存器？中的内容并将其放到光标位置处。这里？可以是一个字母，也可以是一个数字
- ndd：将当前行及其下共n行文本删除，并将所删内容放到1号删除寄存器中。

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	! 外部过滤器	@ 运行宏	# prev ident	\$ 行末	% 括号匹配	^ "软"行首	& 重复:s	* next ident	( 句首	) 下一句首	"soft" bold down	+ 后一行行首
· 跳转到标注	1	2	3	4	5	6	7	8	9	0 一行首	- 前一行行首	= 自动格式化
Q 切换至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	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 左	j 下	k 上	l 右	; 重复	' 跳转到标注的行首	\ 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	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

后面跟随后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

w,e,b命令  
小写(b): quux(foo, bar, baz);  
大写(B): quux(foo, bar, baz);

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: "ay\$ 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), gf: 打开光标处的文件名 (vim only)

原图: www.viemu.com 翻译: fdl (linuxsir)

data:text/html;charset=utf-8,%3Cp%20style%3D%22margin%3A%200px%3B%20padding%3A%200px%3B%20color%3A%20rg... 3/3