

# vim 初步

说明：本文档中，`[key]` 指代键盘上的特殊按键，`<content>` 指代需要输入的内容，单字符命令不作特殊标明。

## 何为 vim

所有的 Unix-like 系统都会内建 vi 文书编辑器，其他的文书编辑器则不一定会存在。目前我们使用比较多的是 vim 编辑器。

vim 是从 vi 发展出来的一个文本编辑器。代码补全、编译及错误跳转等方便编程的功能特别丰富，可以主动的以字体颜色辨别语法的正确性，方便程序设计。在程序员中被广泛使用。连 vim 的[官方网站](#)自己也说 vim 是一个程序开发工具而不是文字处理软件

vim 官方在 Linux 系统中内置了 vim 编辑器的教程，可以使用 `vimtutor zh` 命令进入该教程。教程中特别注明，**要在练习中掌握 vim 的用法**，让我们扔掉鼠标快乐地写代码吧！另外，[菜鸟教程](#)对 vim 编辑器做了相当详细的介绍，可以在此查看 vim 的常用命令。

## vim 架构介绍

vim 作为文本编辑器，最大的特点就是命令式编辑。vim 编辑器有三个模式，将在下文——介绍。

### 一般命令模式

进入 vim 编辑器的指令为 `vim <filename>`，此时进入的就是一般命令模式，窗口显示文件内容，左下角显示文件名，右下角显示当前光标位置（若文件不存在会直接创建文件，同时左下角文件名后提示 [New File]）。这个模式下可以通过各种命令（在右下角显示）对文本执行各种修改操作，但无法引入新的内容。命令在输入结束后会被自动识别并执行。

### 编辑模式

vim 编辑器中直接从键盘读取内容加入文档的模式。此模式下的任何键盘输入都作为普通字符进入文档，不认为是命令。从编辑模式回到一般命令模式需要按下 `Esc` 键；从一般命令模式进入编辑模式需要输入特定命令，包括：

- `i` 从目前光标处进入输入模式；
- `I` 从目前所在行的第一个非空格符处进入输入模式；

- a 从目前光标所在的下一个字符处进入输入模式；
- A 从光标所在行的最后一个字符处进入输入模式；
- o 从目前光标所在的下一行处创建新的行，进入输入模式；
- O 从目前光标所在的上一行处创建新的行，进入输入模式；
- r 进入取代模式，取代光标所在的字符一次；
- R 进入取代模式，一直取代光标所在的字符直到退出编辑模式。

在输入模式和取代模式时，左下角会出现 `--INSERT--` 或 `--REPLACE--` 字样，此时的输入才有效。

## 底线命令模式

是更高级的命令模式，可以执行更加强力的命令，功能更加多元。底线命令模式中的命令输入完毕后需要按 Enter 执行。在一般命令模式输入 `:` 就可以进入底线命令模式，按 Esc 可以退出底线命令模式。

## 一般命令模式常用命令

### 光标移动

- h j k l 或左下上右箭头：光标向左下上右移动一个字符；多次移动可在命令前加入次数。
- [Ctrl] + f/b 或 [PgUp]/[PgDn] 屏幕向下、向上移动一整页。
- [Ctrl] + d/u：屏幕向下、向上半页。
- +/-：光标移动到非空格字符的下/上一行。
- <number>[space]：光标在这一行向右移动若干（由输入决定）字符。
- 0/\$ 或 [Home]/[End]：光标移动至本行的最前/后面字符处。
- <number>[Enter]：光标向下移动若干行。
- H/M/L：光标移动到本屏幕最上方/中央/最下方那一行的第一个字符。
- <number>G：移动到文件第若干行。若不指定number，则移动到最后一行。
- gg：移动到文件第一行。

### 复制、剪切与粘贴

- .：重复上一个动作；
- <number>x/X：向后/前删除若干字符，默认为1；
- <number>dd：剪切光标所在向下的若干整行，默认为1；
- d<number>G：剪切光标到第若干行的所有内容，默认为最后一行；
- d0/d\$：剪切光标到该行第一个/最后一个字符；
- y：复制，用法与 d 完全相同。
- p/P：将已复制的内容在光标下一行/上一行粘贴；

- J：将光标所在行与下一行合并；
- c：重复删除多个数据，如向下删除10行是 10cj；
- u [Ctrl] + r：撤销/重做上一个动作。

## 搜索与替换

- /<word> 和 ?<word>：向光标之下/之上搜索word指代的字符串。
- n/N：同向/反向重复上一个搜寻动作。

## 底线命令模式常用命令

### 替换命令

替换命令： :<range>s/<pattern>/<replacement>/<options>，将word1替换成word2。Arguments 包括：

- <range> 搜寻的范围，默认为当前行。可以是单行的数字 5、表示当行的 . 或末尾行的 \$；也可以使用多行的 <n1>,<n2>，搜寻范围从n1行到n2行；全局搜索可以将两个合并为 %；还可以基于模式写为 <pattern> 或 /<start>/,/end/。
- <pattern> 需要查找的字符或模式。通配符不能直接使用，而要使用正则表达式：.\* 表示任意数量的字符，而 . 表示任意单个字符；/<pattern>\*/ 表示匹配前面的模式一到多次，如 /a\*/ 表示匹配a, aa, aaa等。
- <replacement> 是要被替换成的内容。
  - 若需要替换特定模式的内容，则可以使用正则表达式的分组功能： :%s/\(<pattern1>\)\(<pattern2>\)/\2\1/g 可以将pattern1pattern2替换为 pattern2pattern1，其中前面的两组在后面以 \1 \2 的形式引用。
  - &：相当于 \0，引用整个匹配的内容。
  - \u \l \U \L：分别将替换文本中的字符转为首字母大写/小写或全部大写/小写。
  - \r：插入换行符。
- <options> 是对操作做限定。
  - g：表示执行全局替换操作。
  - c：表示执行替换前显示提示字符让用户确认是否需要替换。
  - i：表示搜索时忽略大小写。
  - n：表示不进行替换而是显示匹配次数。

另外，除了斜杠外，竖线 | 和井号 # 也可以作为分隔符，这适用于查找中本身含有斜杠的情况。

### 储存、离开等

- !：在指令末尾，表示强制

- :w :q :wq : 写入、退出、写入后退出
- :w <filename> : 将文件另存为另一文件, 可用于存新档;
- :r <filename> : 读取另一个文件并将内容加在光标所在行后;
- :n1,n2 w <filename> : 将n1到n2的内容储存为新文件;
- :! <command> : 离开 vim 在系统中执行ls等指令并显示结果;
- :set nu 和 :set nonu : 显示/取消行号。

version 1.1  
April 1st, 06  
翻译:2006-5-21

## 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	\$ 行尾	% 括号 匹配	^ "软" 行首	& 重复 :s	* next ident	( 句首	) 下一 句首	"soft" bol down	+ 后一行 行首
\. 跳转到 标注	1	2	3	4	5	6	7	8	9	0 "硬" 行首	- 前一行 行首	= 自动 <sup>3</sup> 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 撤消 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 <sup>1,3</sup> 行	u 撤消 命令	i 插入 模式	o 分段 (后)	p 粘贴 (后)	[ 杂项	]	杂项
A 在行尾 附加	S 删除行 并插入	D 删除 至行尾	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 <sup>1,3</sup>	f 行内字符 查找	g 附加 命令	h ←	j ↓	k ↑	l →	;	重复 u/T/f/F	' 跳转到标 注的行首	\. 未用!
Z 退出 <sup>4</sup>	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 <sup>3</sup>	> 缩进 <sup>3</sup>	?	向前 搜索		
Z 附加 命令 <sup>5</sup>	x 删除 (字符)	c 修改 <sup>1,3</sup>	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

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](http://www.viemu.com) 翻译: fdl (linuxsir)