

vim-map

vim-map

软件版本：
ubuntu10.04
Linux version 2.6.32-42-generic
VIM – Vi IMproved 7.2
目录：

- 1. 简介
- 2. 映射与运行模式的关系
- 3. 例子
- 4. 其他

1. 简介

如果想让 vim 成为你手中的利器，vim 的 map 功能就不得不学，它用于键映射、缩写和自定义命令。由于 map 命令有很多，不同的命令作用与不同的模式之下，文章主要是想分析一下各个命令与模式之间的关系。

2. 映射与运行模式的关系

映射是否起作用的一个关键因数是当前 vim 处于哪种模式之下。map 与六种 vim 模式相关（vim 有多少种工作模式请参考 :help vim-modes）：

- (o) 普通模式：输入命令时。
- (x) 可视模式：可视区域高亮并输入命令时。
- (s) 选择模式：类似于可视模式，但键入的字符对选择区进行替换。
- (o) 操作符等待模式：操作符等待中（"d"，"y"，"c" 等等之后）。
- (i) 插入模式：也用于替换模式。
- (c) 命令行模式：输入 ":" 或 "/" 命令时。

了解到 map 是与 vim 的工作模式相关之后，这时候再看一下支持嵌套和递归的 map 命令命名有什么特别之处（:help map）：

命令	左边	右边	模式	跳跃点
:map	{lhs}	{rhs}	mapmode-nvo	*:map*
:nm[ap]	{lhs}	{rhs}	mapmode-n	*:nm* *:nmap*
:vm[ap]	{lhs}	{rhs}	mapmode-v	*:vm* *:vmap*
:xm[ap]	{lhs}	{rhs}	mapmode-x	*:xm* *:xmap*
:smap	{lhs}	{rhs}	mapmode-s	*:smap*
:om[ap]	{lhs}	{rhs}	mapmode-o	*:om* *:omap*
:map!	{lhs}	{rhs}	mapmode-ic	*:map!*
:im[ap]	{lhs}	{rhs}	mapmode-i	*:im* *:imap*
:lm[ap]	{lhs}	{rhs}	mapmode-l	*:lm* *:lmap*
:cm[ap]	{lhs}	{rhs}	mapmode-c	*:cm* *:cmap*

不难发现 map 命令的命名与它所支持的 vim 模式相关。下面再列出具体的关系：

- 字 符 模 式
- <Space> 普通、可视、选择和操作符等待
 - n 普通
 - v 可视和选择
 - s 选择
 - x 可视
 - o 操作符等待
 - ! 插入和命令行
 - i 插入
 - l 插入、命令行和 Lang-Arg 模式的 ":lmap" 映射

c 命令行

总结，map 命令的首字母表明它作用于何中 vim 模式之下。

3. 例子

我以一些简单的命令演示一下 map 命令与 vim 模式之间的关系。

1) 普通模式

输入一下命令：

```
:nmap ;y "+y
```

这样，我们就可以使用 ;y 替代 "+y 将内容复制到全局缓冲中。但是如果你首先使用<C-V>选中一块数据，然后按 ;y，结果没有将数据复制到全局缓冲中，只能复制到局部缓冲。这是因为 nmap 只在普通模式下起作用。

2) 可视模式

输入下面这条命令之后，上面提到的问题就不复存在了。

```
:xmap :y "+y
```

我们再来看一个好玩的。

3) 选择模式

如果你不知道怎么进入选择模式，请参考 :help select 。我觉得一般情况下使用选择模式都是首先进入可视模式，然后键入 <CTRL-G> 进入选择模式。因为当你处在选择模式下，键盘的输入将替换选中区域。

```
:smap ;y "+y
```

看上面的命令与之前的例子大同小异吧。但效果却截然不同。操作过程：

按 v，进入可是模式，然后左右上下移动选中一片区域；

按<CTRL-G>进入选择模式；

按 ;y 之后我们看到： "+y 。

为什么他没有帮我们实现复制的功能？因为当前处于选择模式，输入的键值将被直接替换选中区域，而我们输入的 ;y 被映射成 "+y 。

4) <Space>

<Space> 支持的的模式包括 nvo，其中 v 包括 x 和 s 。所以用 <Space>，也就是 :map 命令，就足够了，不需要重复键入那么多吃同样的映射命令。

```
map ;y "+y
map ;p "+p
```

用这个命令的另一个好处就是即使在选择模式下，;y 仍然完成复制操作。

5) 操作等待模式

有必要说一下什么叫操作等待模式吗？它其实跟普通模式很接近，但是出现在输入操作符之后。例如你在普通模式下输入 y 则会进入操作等待模式一会儿，它在等待 y 的组合键，例如 w 。所以，我们可以输入一下命令做一下认证：

```
:omap [ w
```

当我们在普通模式下输入 y[的作用就等同与 yw 。

6) 插入模式

加入下面这几条命令，在输入时，如果需要左右移动光标，我们也不需要把右手移动到方向键上面去了。

```
:imap <C-G>l <Right>
:imap <C-G>h <Left>
:imap <C-G><C-L> <C-Right>
:imap <C-G><C-H> <C-Left>
```

在输入模式下按下 <CTRL-G> 再按 l，光标右移。<C-G> + <C-L> 光标右移一个字，相当于普通模式下的 w 。

7) 命令模式

按葫芦画瓢，在命令模式下也可以这样干。

```
:cmap <C-G>l <Right>
:cmap <C-G>h <Left>
:cmap <C-G><C-L> <C-Right>
:cmap <C-G><C-H> <C-Left>
```

8) !

既然命令模式和插入模式的映射是一样的，就没有必要敲那么多命令了，用 map! 就好了。

```
:map! <C-G>l <Right>
:map! <C-G>h <Left>
:map! <C-G><C-L> <C-Right>
:map! <C-G><C-H> <C-Left>
```

4. 其他

有时间再补。

分类: [linux,vim](#)

标签: [vim](#), [map](#)

好文要顶

关注我

收藏该文

[Eddy_He](#)
[关注 - 8](#)
[粉丝 - 11](#)
[+加关注](#)

0
推荐

0
反对

« 上一篇: [vim-project](#)
» 下一篇: [vim-man](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#)网站首页。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库



- 最新IT新闻:
- 余承东谈消费升级和消费者业务：华为不仅是技术、踏实的“理工男”
 - 人人视频宣布转型海外短视频内容社区
 - 马云：假货既是阿里之痛，更是中国之痛
 - 苹果正起诉Swatch的“Tick different”商标涉侵权
 - 淘汰锂电 中国教授发明磷纳米电池：手机续航增7倍
- » 更多新闻...



- 最新知识库文章:
- [如何打好前端游击战](#)
 - [技术文章的阅读姿势](#)
 - [马拉松式学习与技术人员的成长性](#)
 - [程序员的“认知失调”](#)
 - [为什么有的人工作多年还是老样子](#)
- » [更多知识库文章...](#)

昵称: [Eddy_He](#)
园龄: [5年1个月](#)
粉丝: [11](#)
关注: [8](#)
[+加关注](#)

2012年9月						
<	日	一	二	三	四	五
	26	27	28	29	30	31
	2	3	4	5	6	7
	9	10	11	12	13	14
	16	17	18	19	20	21
	23	24	25	26	27	28
	30	1	2	3	4	5

搜索

常用链接

- [我的随笔](#)
- [我的评论](#)
- [我的参与](#)
- [最新评论](#)
- [我的标签](#)

我的标签

- [vim](#)(16)
- [git](#)(6)
- [ubuntu](#)(5)
- [linux](#)(5)
- [android](#)(3)
- [cache](#)(2)
- [eclipse](#)(2)
- [jni](#)(2)
- [file](#)(1)
- [ubuntu ibus 中文](#)(1)
- [更多](#)

随笔分类

- [Android](#)(2)
- [C](#)(2)
- [git](#)(6)
- [JAVA](#)(1)
- [linux](#)(11)
- [office](#)(1)
- [shell](#)(2)
- [ubuntu](#)(10)
- [vim](#)(23)
- [逻辑推理](#)(1)

随笔档案

- [2013年5月](#) (1)
- [2012年11月](#) (1)

- 2012年9月 (13)
- 2012年8月 (3)
- 2012年7月 (5)
- 2012年6月 (3)
- 2012年5月 (4)
- 2012年4月 (2)
- 2012年3月 (23)

文章分类

- C
- git
- linux
- office
- shell
- ubuntu

最新评论

1. Re:vim-project

请问下，图中的vim配色方案是什么啊？谢谢！

--山本二十八

2. Re:git-分支管理

感谢分享

--FMN

3. Re:vim-echofunc

mark学习了.

--mfslog

阅读排行榜

- 1. git-commit(70785)
- 2. E: Sub-process /usr/bin/dpkg returned an error code (1) 解决方案(27890)
- 3. git-忽略指定文件(20115)
- 4. Android与JNI(二) (7841)
- 5. Android与JNI(一) (5708)

评论排行榜

- 1. git-分支管理(1)
- 2. vim-echofunc(1)
- 3. vim-project(1)

推荐排行榜

- 1. git-忽略指定文件(3)
- 2. git-分支管理(1)
- 3. git-commit(1)
- 4. Android与JNI(一) (1)