

当前位置：[首页](#) > [入门](#) > [Linux命令](#) > Vim实用技巧

Vim实用技巧

发布时间：2014-10-16 16:38:29 编辑：AHLinux.com

vim是Vi IMproved，是编辑器Vi的一个加强版，一个极其强大并符合IT工程师（程序员、运维）习惯的编辑器。

（一）初级个性化配置你的vim

1、vim是什么？

vim是Vi IMproved，是编辑器Vi的一个加强版，一个极其强大并符合IT工程师（程序员、运维）习惯的编辑器。如果你是一名职业的SE，那么一定在寻找一款出色的能够自由定制、满足灵活编辑功能的编辑器。那么答案，就是vim或者Emacs。而这一套连续的博文，就为您介绍vim编辑器。至于另一款强大的编辑器Emacs，我们会在今后的一个系列博文中看到。

2、配置文件在哪？

Windows系统的配置文件是vim安装目录下的vimrc文件。

Linux系统，RHEL和CentOS的配置文件是/etc/vimrc，Debian和Ubuntu的配置文件是/usr/share/vim/vimrc。

Mac OS X的配置文件是/usr/share/vim/vimrc。

3、vimRC中的RC是什么意思？

为什么把这么个没有意义的问题也列出来？原因很简单，就是我在接触vim之初第二个想问的问题就是这个（第一个就是上面的“1”）。一般在运行某个软件、环境或工具之初，要初始执行的一些命令，称为Run Commands，即RC。根据Wikipedia.org的记载，RC缩写的用法源自MIT的相容分时操作系统（Compatible Time-Sharing System，CTSS）[参考1]。所以以后如果你看到了SystemRC或者ScreenRC，也就知其缘由了。

4、三种基本模式

在《大家来学vim》中提到vim有6+5种模式，但除了细分程度不同外，实际上常用的只有3种模式：Normal Mode、Insert Mode和Command-line Mode。

从Normal Mode进入Insert Mode，可以按i、I、a、A、o、O、r、R即可。其中i和I表示插入（Insert），r和R表示替换（replace）。

从Insert Mode进入Normal Mode，只需要按ESC键即可。

从Normal Mode进入Command-line Mode，可以按“:”、“/”、“?”。其中“:”表示输入vim命令，“/”表示后面输入要向下搜索的字符串，“?”表示后面输入要向上搜索字符串。

从Command-line Mode进入Normal Mode，只需要按ESC键即可了。

你可以把Normal Mode理解为常态，进入其他任何一种模式，需要某种特殊方式，而从其他特殊模式回到Normal Mode，就按ESC键即可。

5、vim中那些最最常用的配置

当你尝试去Google一些其他人的vimrc配置时，你一定会找到一篇叫做《The ultimate vim configuration》的文章，不过它的作者，Amix后来在他的博客上提到[参考2]，这份在google搜索vimrc会排在前十的vim配置文件，如今已经过时了，所以他提供了一些更新信息。

(1) 颜色设置

```
syntax on      "开启代码高亮
syntax off    "关闭代码高亮
syntax enable "开启代码高亮
```

(2) 搜索设置

```
set hlsearch "开启搜索高亮
set nohlsearch "关闭搜索高亮
set incsearch "输入搜索字符串的同时进行搜索
set ignorecase "搜索时忽略大小写
```

(3) 用户界面

```
set showmode "开启模式显示
set ruler    "开启光标位置提示
set number   "显示行号
```

```
set nonu          "不显示行号
set cursorline    "强调光标所在行
set cmdheight=1   "命令部分高度为1
```

(4) 编辑辅助配置

```
set autoindent    "自动缩进
set noautoindent  "不自动缩进
set smartindent   "智能缩进
set autoread      "当文件在外部被改变时，vim自动更新载入
set showmatch     "显示匹配的括号
```

参考：

Run Commands , http://en.wikipedia.org/wiki/Run_commands

The Ultimate vim Configuration(vimRC) , <http://amix.dk/blog/post/19486>

(二) 常用的状态切换按键

1、Normal Mode -> Insert Mode

i 小写字母i，在光标位置插入
a 小写字母a，在光标的下一个位置插入
I 大写字母I，在光标所在行的第一个非空格处插入
A 大写字母A，在光标所在行的最后一个字符处插入
o 小写字母o，在光标所在行的下一行处插入新行
O 大写字母O，在光标所在行的上一行处插入新航
r 小写字母r，替换光标所在处的字符一次
R 大写字母R，持续替换光标所在处的字符，直到按下ESC

2、Normal Mode -> Command-line Mode

:w 保存文件
:w! 强制保存文件（前提是用户有修改文件访问权限的权限）
:q 退出缓冲区
:q! 强制退出缓冲区而不保存
:wq 保存文件并退出缓冲区

:wq! 强制保存文件并退出缓冲区（前提是用户有修改文件访问权限的权限）

:w 另存为名为filename文件

:n1,n2 w 将n1行到n2行的数据另存为名为filename文件

:x 如果文件有更改，则保存后退出。否则直接退出。

3、Insert Mode -> Normal Mode

按下ESC键

4、Command-line -> Normal Mode

按下ESC键

（三）常用光标移动按键

1、光标字符操作

j 向下

k 向上

h 向左

l 向右

\$ 光标移动到行尾（End），注意要按Shift键

O 光标移动到行首（Home）

^ 光标移动到行首第一个非空白字符（Home），注意要按Shift键

2、光标词操作

w 光标移动到后一词的词首

W 光标移动到后一词的词首且忽略标点符号

e 光标移动到后一词的词尾

E 光标移动到后一词的词尾且忽略标点符号

b 光标移动到前一词的词首

B 光标移动到前一词的词首且忽略标点符号

3、光标句操作

) 光标移动到后一句的句首

(光标移动到前一句的句首

% 配合 "(" 和 ")" 使用，寻找相匹配的另一半

4、光标行操作

G 光标移动到文档的最后一行的第一个非空白字符

nG 光标移动到文档的第n行，相当于" :n"

gg 光标移动到文档的第1行的第一个非空白字符，相当于" 1G"，也相当于" :1"

光标向下移动N行

5、光标段操作

} 光标移动到下一段的段首

{ 光标移动到上一段的段首

% 配合 "(" 和 ")" 使用，寻找相匹配的另一半

6、光标页操作

Ctrl+f 向下翻页 (Pagedown)

Ctrl+b 向上翻页 (Pageup)

Ctrl+d 向下翻半页

Ctrl+u 向上翻半页

H 光标移动到目前显示页的第一行

M 光标移动到目前显示页的中间行

L 光标移动到目前显示页的最后一行

7、光标自由操作

Ctrl+o 回到光标的上一位置

(四) 常用编辑操作按键

1、删除操作 (delete)

dd 删除光标所在行

ndd 删除从光标所在行开始，向下的n行

d1G 删除从光标所在行开始，到第一行的所有行

dG 删除从光标所在行开始，到最后一行的所有行

d\$ 删除从光标所在位置，到行尾的所有字符

d0 删除从光标所在位置，到行首的所有字符

2、复制操作 (yank)

- yy 复制光标所在行
- n yy 复制从光标所在行开始，向下的n行
- y1G 复制从光标所在行开始，到第一行的所有行
- yG 复制从光标所在行开始，到最后一行的所有行
- y\$ 复制从光标所在位置，到行尾的所有字符
- y0 复制从光标所在位置，到行首的所有字符

3、粘贴操作 (paste)

- p 小写字母p，粘贴剪贴板中的数据，从光标所在行的下一行开始
- P 大写字母P，粘贴剪贴板中的数据，从光标所在行的上一行开始

4、撤销与重做操作 (undo , redo)

- u (Undo) 撤销上一个操作
- Ctrl+r (Redo) 重做上一个操作

5、重复操作

- . 重复上一操作

6、替换操作 (replace)

- r 替换光标所在处字符
- R 进入替换模式，直至按ESC退出
- cc 替换光标所在行
- cw 替换光标所在的英文单词
- ~ 转换大小写

7、排版操作

- :le 光标所在行左对齐
- :ri 光标所在行右对齐
- :ce 光标所在行居中

(五) 常用多缓冲区操作按键

1、多文件编辑模式

argument list模式，就是在打开vim编辑器时，携带多个文件路径参数。

buffer list模式，就是在进入vim编辑器后，打开多个缓冲区进行编辑。

2、单一缓冲区打开多个文件

:files 显示目前打开了哪些文件

:n 切换到下一个缓冲区内容

:N 切换到上一个缓冲区内容

:2n 切换到下下个缓冲区内容

:bn 下一个缓冲区内容 (buffer next)

:bp 上一个缓冲区内容 (buffer previous)

3、多个缓冲区打开多个文件

:sp [filename] 打开一个新的缓冲区，如果有filename则内容为该文件，否则为当前文件

Ctrl+w n 新建一个缓冲区

Ctrl+w q 退出光标所在的缓冲区

Ctrl+w j 光标移动到下一缓冲区

Ctrl+w k 光标移动到上一缓冲区

Ctrl+w l 光标移动到右一缓冲区

Ctrl+w h 光标移动到左一缓冲区

Ctrl+w v 左右切割窗口新建缓冲区

Ctrl+w s 上下切割窗口新建缓冲区

Ctrl+w o 使光标所在缓冲区最大化，其他缓冲区隐藏

(六) 常用搜索与书签操作快捷键

1. 搜索字符串

/string 向下搜索字符串 “string” ?string 向上搜索字符串 “string”

2. 重复上次搜索

n 根据上次搜索条件找到下一个匹配的字符串 N 根据上次搜索条件找到上一个匹配的字符串

3. 搜索单词

* 向下搜索光标所在处的单词 (完全匹配) # 向上搜索光标所在处的单词 (完全匹配) g* 向下搜索光标所在处的单

词（部分匹配）g# 向上搜索光标所在处的单词（部分匹配）

4. 标记书签（mark）

ma a为小写字母，为光标所在处设定文档书签amA A为大写字母，为光标所在处设定全局书签A

5. 使用书签（mark）

`a 到文档书签a处，Tab键上方'a 到文档书签a所在行行首处，Enter键左边'A 到全局书签A处，Tab键上方'A 到全局书签A所在行行首处，Enter键左边`n 如果n=0，缓冲区将打开上一次的文档，且光标在上次编辑最后的位置，1-9以此类推'n 如果n=0，缓冲区将打开上一次的文档，且光标在上次编辑最后的位置所在行的行首，1-9以此类推

6. 查看书签列表

:marks 查看当前所有书签

（7）关于加密

1、加密你的文本文件

当你用vim打开或新建一个文件时，如果加上-x参数，即：

```
vim -x filename
```

那么vim会提示你：

Enter encryption key:

当你输入了密码后，vim会提示你重复输入以确认：

Enter same key again:

你在设置密码的时候，是以*来显示的，非明文。当下次有人（包括你自己）打开该文件的时候，vim就会提示：

Need encryption key for "main.cpp"

Enter encryption key:

这时你只要输入你之前所设置的密码就可以了。如果你输入的密码是错误的，vim并不会提示你密码错误，而是显示一堆无意义的垃圾代码。

2、解除加密与重设密码

打开你已经加密的文件，然后输入：

```
:set key=
```


这样，你的密码就被设置为空了，今后进入就不再需要输入密码了。而如果你向重设密码的话，你可以输入：

:X

注意是个大写字母X。这时vim会提示你，只要反复输入两次新的密码就可以了：

Enter encryption key: *****

Enter sanme key again: *****

3、加密引起的问题

vim的加密功能，可以说是问题一大堆。比如：

（1）如果密码输入错误，vim不会拒绝访问者编辑文件。在这种显示乱码的情况下编辑文件并保存，整个文件就会变得混乱不堪，无法辨识。今后即使使用正确的密码打开文件，也看不到正确的文档了。

（2）如果对一个需要编译的源代码文件进行了加密，则无法通过编译器的语法检查。

（3）交换文件是无法加密的。这样引起什么问题？问题就是如果其他用户想办法读取到了你的交换文件，就可以绕过解密环节，你的加密也就无意义了。解决的办法就是，不使用交换文件。方法有三个，一是修改你的vimrc文件（vim配置文件）：

```
set noswapfile
```

二是在用vim打开文件之初就输入命令：

```
vim -x -n filename
```

（注意-x是加密参数，与我们所说的swap没有关系，不要搞混）

三是在进入vim编辑该文件后，输入：

```
:setlocal noswapfile
```

不过不使用交换文件的坏处就是，你可能会面临数据丢失而无法找回的悲剧。所以又想用加密，又想不丢失文件的朋友们，就勤按“:w”吧。

本文来自：爱好Linux技术网

本文链接：<http://www.ahlinux.com/start/cmd/2829.html>



相关关键词搜索：Vim实用技巧 vim

上一篇: linux解压-tar命令详解1

下一篇: vi删除空行的方法

相关文章

- Vim设置编辑Python文件时自动缩进为4个空格的方法 2014-10-16
- vim退出后恢复终端内容的方法 2014-10-16
- VIM注释插件及相关英文解释 2014-10-16
- 配置 vim 显示汉字 2014-10-16
- vim学习进阶：better、faster、stronger 2014-10-16
- vim技巧之翻页 2014-10-16
- vim命令教程(最全面的vim教程)专题页 2014-10-16
- Vim替换字符串的一些技巧总结 2014-10-16
- vim配置文件(经典实用，值得珍藏) 2014-10-16
- vim技巧之添加程序注释 2014-10-16
- VIM使用正则表达式搜索字符串的方法详解 2014-10-16
- 解决vi/vim中粘贴时行首出现很多缩进和空格的问题 2014-10-16

关键词：



震惊

不解

愤怒

杯具

无聊

高兴

支持

超赞

相关评论

[关于本站](#) | [网站地图](#) | [联系我们](#) | [免责声明](#) | [友情链接](#)

皖ICP备09020804号

CopyRight © 2009-2014 **爱好Linux技术网** AHLinux.com , All Rights Reserved.