

### 一、什么是vim?

Linux的一个重要设计原则就是信息存储在基于文本的文件中，因此使用文本编辑器来编辑配置文件非常重要与常见。文本文件包括无格式文件（内有多行相似的信息，如/etc中的配置文件）和可扩展标记语言（XML）文件（通过文本标记定义数据结构，如/etc和/usr中的应用配置文件）。文本文件的优点是它们可以在系统之间移动或者共享，无需转换，而且可以通过任何简单文本编辑器进行查看和编辑。vim就是其中一种文本编辑器。

#### 1、为何要掌握vim?

那么问题来了，文本编辑器那么多，为什么我们一定要掌握vim编辑器的使用方法？每个系统管理员对于文本编辑器都有偏好，一些人会选择gedit，另一些则喜欢nano，设置还有人选用emacs。即使已经有选用的编辑器，也务必要掌握vim或vi的基本知识，原因很简单：vim是随Linux和UNIX系统分发的vi编辑器的改进版本。无论操作的系统是什么，此款编辑器都能可靠地安装。

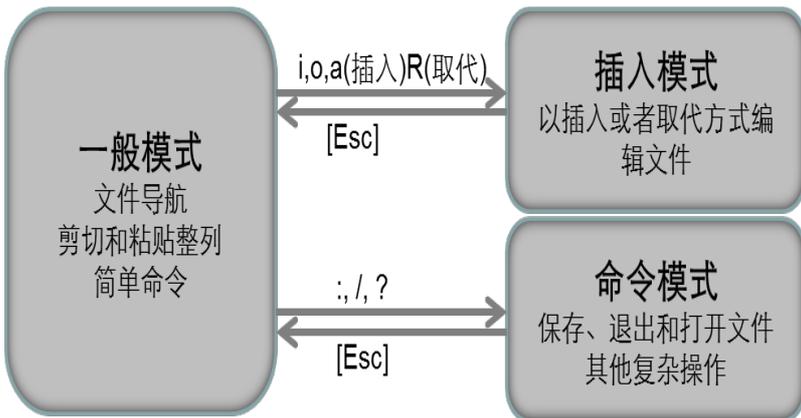
#### 2、vim和vi的区别?

vim是vi的高级版本，具有vi的所有文字处理功能，且具有许多原版vi中没有的功能，但基本上依然保持向后的兼容性。新功能中包含了广受欢迎的选项，如语法突出显示、补全模式和拼写检查等。

说明：Linux操作系统上调用vi命令时，有时候发现执行的命令是vim。这是shell启动时通过/etc/profile.d/vim.sh中设置的别名完成的。在需要比较新的功能时，建议始终执行vim，而不要依赖不一定可用的别名。用户通常必须以root身份操作时尤其建议这么做。

#### 3、vim的三种模式

**开始学习喽！** vim并不是最容易学习的编辑器，其部分原因是vim中的所有命令都以速度和效率为目标而设计，不易记忆，另一部分原因在于vim是模拟编辑器。模拟编辑器意味着特定命令和按键操作的功能取决于活动状态的模式是什么。Vim有三个主要工作模式详细见下。



从上图可以看出一般模式可以与插入模式及命令模式切换，但是插入模式和命令模式之间不可切换。

三种模式详细解释见下：

- | 一般模式是vim打开文件的默认模式，用于文件导航、剪切和粘贴以及简单命令。撤销、恢复和其他操作也从此模式中执行。可使用“上下左右”案件来移动光标。
- | 插入模式用于常规文本编辑。替换模式是插入模式的一种变体，可以替换而不是插入文本。
- | 命令模式用于保存、退出和打开文件，以及搜索、替换和其他更为复杂的操作。从此模式中，可以将程序的输入输出插入到当前文件中，以及配置vim等的功能。使用ex可以实现的所有功能都可以从此模式执行。

了解到这个基础知识，你就可以开始简单的使用vim进行文本编辑了！

### 二、vim基本工作流程

#### 1、打开文本文件

使用vim文本编辑器开始编辑文件有一个最简易的做法，就是将其指定为命令行上的一个参数。比如我们要编辑/etc目录下的passwd文件，可以执行以下命令：

```
root@HZ-CAS01-CVK01:~# vim /etc/passwd
```

说明：

如果尝试打开编辑的文件并不存在，但是却存在指定的目录，vim将通知您正在编辑一个[新文件]，并将在首次进行保存时创建该文件。

例如您想编辑的是/etc目录下的passwd文件，结果输入命令的时候输成了password文件，如下所示：

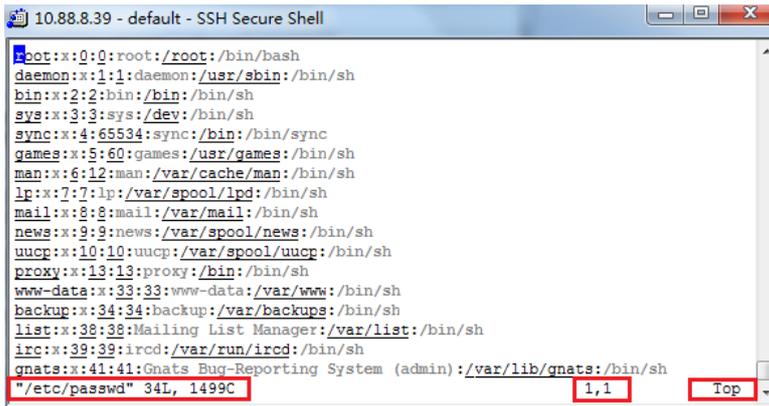
```
root@HZ-CAS01-CVK01:~# vim /etc/password
```

实际/etc目录下没有该文件，但是/etc目录是可达的。所以执行完命令后，在左下角会提示“/etc/passwd [New File]”。此时如果退出编辑时候选择保存则该文件就会被创建，如果直接退出则不会创建。

```
~/
"/etc/password" [New File]
```

打开文件后，vim将在一般模式中启动。此时可以在屏幕左下角看到有关已打开的文件信息（文件名、

行数、字符数），并在右下角看到当前的光标位置（行、字符）以及正在显示文件的哪个部分（All表示全部，Top表示文件的前几行，Bot表示文件底部，或者显示百分比来表示您所处的文件位置）。最下方线条在vim术语中成为标尺。



## 2. 编辑文本

如果之前使用过vi或者vim，可能会发现一般模式中多数键并不会完全执行预期的操作。比如敲“i”键后文本中不会键入i字母。这是因为在一般模式中，键的映射不是为了插入所按下的字符，而是为了执行诸如光标的移动、复制和粘贴等操作的命令。在插入模式下可正常插入或者取代方式编辑文件。那么问题来了，如何进入插入模式？系统提供了可用的命令能够从一般模式切换到插入模式，且每个命令在键盘上都分配有一个不同的键：

按键	执行结果
i	切换到插入模式，并在当前光标位置之前开始插入（插入）
I	将光标移至当前行的开头位置，并切换至插入模式
a	切换到插入模式，并在当前光标位置之后开始插入（附加）
A	将光标移至当前行的结尾位置，并切换至插入模式
R	从光标下的字符开始，切换到插入模式。处于插入模式时，不会插入文本，此时输入的每个字符均将替换掉当前文档的字符。（vim和vi也具有更多更强大的替换命令， <a href="#">详见第三章第8节</a> ）
o	在当前行的下方新打开一个行，并切换到插入模式
O	在当前行的上方新打开一个行，并切换到插入模式

每次处于插入模式时，标尺将会显示—INSERT—或者—REPLACE--。要返回至命令模式，可以按Esc。详细三种模式切换关系见“vim的三种工作模式”小节。

在一般模式中移动光标可以使用多个按键，这些按键执行效果见下，移动好光标后再进入插入模式进行编辑会非常方便：

按键	在一般模式中光标移动结果
h	光标向左移动一个位置
l	光标向右移动一个位置
j	光标向下移动一行
k	光标向上移动一行
^	移至当前行的开头位置
\$	移至当前行的结尾位置
gg	移至文档的第一行
G	移至文档的最后一行

说明：Esc将会取消当前命令，或者返回至一般模式。常见的做法是按两下Esc(或者按更多下)，以确保返回至命令模式。

## 3. 保存文本文件

在vim中保存文件时通过在命令模式中实现的。要进入命令模式，可在一般模式中按：（冒号）并等待键入命令（详见见第一张的第3节）。通过按Enter可完成命令。

以下简要列出从命令模式保存并推出当前文件时可使用的命令。

按键	执行结果
:wq	保存并退出当前文件
:w	保存当前文件并留在编辑器中
:w!	若文本为只读模式，强制保存
:q	退出当前文件（仅在没有未保存更改的情况下）
:q!	退出当前文件，忽略任何未保存的修改

简要概括来说，w用于保存（写入），q用于退出，！用于强制执行操作。

说明：表中以及后续表格并非所有命令，仅列出常用命令。

## 4. 帮助

vim在编辑器中提供内容广泛的在线帮助。在一般模式下键入:help将启动第一个屏幕，其中包括导航帮助所需的帮助。

通过在命令模式中键入:help subject，可获取有关特定主题的帮助。

帮助屏幕在新的拆分窗口中打开并可使用:q关闭。要了解更多拆分窗口的更多所信息,使用:help window。

### 三、使用vim编辑

#### 1. 移动快捷方式 (一般模式)

除了用户可以在一般模式中执行单调而老套的单字符/单行光标移动外,也有一些高级移动命令,帮助用户更加高效地在文本中导航。这些快捷方式可以使光标按照单词、句子或段落移动。请注意,与常规的光标移动不同,这些命令仅可以在一般模式中使用,而不能在插入模式中使用。

按键	执行结果
w	将光标移到下一个单词的开头 (W包含标点符号)
b	将光标移到上一个单词的开头 (B包含标点符号)
(	将光标移到当前或上一句子的开头
)	将光标移到下一句子的开头
{	将光标移到当前/上一段落的开头
}	将光标移到下一段落的开头

#### 2. 替换文本 (一般模式)

vim允许用户通过“change”命令替换大量 (和少量) 文本。

“change”命令的使用方法为:按c键,后面加上光标移动;例如cw可以将当前光标位置更改到当前单词的末尾。要替换的文本被删除 (防止到未命名寄存器中),vim也会切换到插入模式。

可以通过一些快捷方式提高编辑的效率:

按键	执行结果
cc	开始以行范围的方式替换,即替换一整行 (或者带上数字前缀时替换多行)。同样的技巧也是用于许多其他命令 (如删除)。
i或者a做前缀	以选择inner或a版本的移动。例如,ciw将替换整个当前单词,而不仅仅是从当前光标位置起,而caw则执行相同的功能,但包含周围的任何空白区。
c\$或者C	替换到行末为止
r	替换光标处的字符
~	改变光标处字符的大小写

#### 3. 删除文本 (一般模式)

用于删除文本的命令是d,而且对更改文本有效的所有相同移动也适用于删除,包括D可以删除光标位置到行末的内容。要仅删除光标处的字符,可使用x。具体删除文本的操作方式和替换文本相同,如删除光标所在整行可输入dd,如果要删除光标所在行的内容可输入D。详细见下:

按键	执行结果
x	向后删除一个字符
X	向前删除一个字符
dd	删除光标所在的那一行
ndd	删除光标所在的向下n行 (n为数字)
D	删除光标所在行的内容 (不删除行)
d1G	删除光标所在行到第一行的所有数据
dG	删除光标所在行到末行的所有数据

#### 4. 复制和粘贴 (一般模式)

vim用于描述和粘贴操作的术语与大多数人目前熟悉的稍有不同。复制操作成为拖拉,而粘贴则叫做放置。这体现在分配给这些操作的键盘命令上:拖拉是y键加上移动,而放置操作则通过p和P执行。

拖拉操作遵循与替换和删除操作相同的通用方案:用户可以选择键入要重复某一操作的次数,后面加上y,再加上一种移动。例如,5yaw将复制当前的单词,以及后面四个单词 (共五个)。按yy将拖拉一整行等等。

放置 (黏贴) 通过p和P命令执行;小写p将内容放置到当前光标位置的后面 (如果粘贴的是行范围的数据,则放在当前行的下面),而大写P则放置到当前光标位置的前面或者当前行上方。与所有命令一样,放置命令可以加上寄存器黏贴次数作为前缀。

##### 寄存器:

vim不是仅提供一个供复制与粘贴使用的剪贴板,而是提供26个命名寄存器,以及多个特殊用途的寄存器。拥有多个寄存器可以让用户更加高效的执行剪切和粘贴,而不必担心丢失数据,或者过多地移动光标。如果未指定要使用的寄存器,则将使用“未命名”寄存器。常规寄存器成为a到z,通过在命令的技术和实际命令之间放入“registername”来选择。例如,要复制当前行和后面两行到t寄存器中,用户可以使用3\*tyy命令。

要放置命名寄存器中的内容,只需在放置命令之前加上“registername”。例如,sp将在光标位置后面放置s寄存器中的内容。

重要:每次使用命名寄存器时,未命名寄存器也会更新。

删除和更改操作也可以加上寄存器选择作为前缀。未指定寄存器时,将仅使用未命名寄存器。当使用寄存器的大写版本时,被剪切或拖拉的文本将附加到该寄存器,而不是覆盖它。

##### 特殊寄存器:

有10个数字编号的寄存器,分别为“0”到“9”。寄存器“0”将始终拥有最近拖拉的文本的副本,而寄存器“1”则具有最近删除的文本的副本。当新的文本被更改或删除时,“1”的内容将移到“2”中,“2”的内容将移到“3”中。

重要:与命名寄存器不同,数字编号寄存器的内容不会在会话之间保存。

## 5. 搜索 (一般模式)

在当前文档中搜索可以通过2种方法启动,按/键从光标处向后搜索,或者按?键从当前光标位置向前搜索。当进入搜索模式后,可以键入要搜索的正则表达式,然后按Enter键调到第一个匹配项(若有)。

按键	执行结果
/	从光标处向后搜索:在"/"后输入要搜索的内容,按Enter键开始搜索
?	从光标处向前搜索:在"?"后输入要搜索的内容,按Enter键开始搜索

## 6. 搜索和替换 (命令模式)

vim中的搜索和替换通过命令模式实施,其使用的语法与用户通过sed搜索和替换时使用的相同,包括使用正则表达式进行搜索的功能:

ranges/pattern/string/flags

range可以是行号(42)、行号范围(1,7表示第1-7行)、搜索条件(/README.txt)、%(当前文档中的所有行;搜索和替换通常仅针对当前行操作),或者'<,>'(当前的可视选择)。

两个最为常见的是flags是:g(替换一行中多个位置的pattern)和i(使当前的搜索区分大小写)。

搜索和替换案例

例如,要在所有行中搜索每个位置上的“cat”一词并将它替换成“dog”,不区分大小写,但仅针对完整的词语而不是诸如“catalog”中的一部分,可以使用下列命令:

```
: %s<cat>/dog/gi
```

## 7. 文本显示和取消显示行号 (命令模式)

vim编辑文本过程中显示行号,在编辑过程中非常方便。

按键	执行结果
:set nu	显示行号
:set nonu	取消显示行号

## 8. 撤销和恢复 (一般模式)

为了允许人为误差的存在,vim配备了撤销/恢复机制。只要在一般模式中按u即可撤销最近一个操作i。如果撤销了过多的操作,按Ctrl+r即可恢复最近一次撤销。

奖励诀窍:从一般模式中按.(句点)将对当前行恢复最近一个编辑操作。这可以用于对多行轻松执行同一编辑操作。

参考:vim(1)man page和vim内置帮助。

## 四、摘要小结

vim文本编辑器

vim有三个主要模式:

- | 一般模式用于文件导航和简单命令
- | 插入模式用于常规文本编辑
- | 命令模式用于保存、推出和执行更为复杂的制定

vim基本工作流程

- | 光标键和hikl均可用于移动光标
- | Escape用于退出当前命令或者模式,按两下可始终在一般模式中终止
- | :w用于保存,:q用于退出,:wq用于保存和退出

使用vim编辑

- | 快速光标命令:wb(){}
- | c进入更改模式
- | d和y用于剪切和复制,p用于粘贴

## 五、实战练习

实验要求:

- 1、在/vms路径下创建新文件vim-practice.txt
- 2、插入以下文本  
This is my vim practice.txt  
There are many like it,but this one is mine.
- 3、在底部插入新行,并键入以下内容:  
More lines,I want more line!
- 4、撤销之前的编辑
- 5、恢复(即撤销刚才的撤销操作)最近的编辑
- 6、保存并退出该文件。

实验操作指导:

- 1、创建文件:在vim中打开(新)文件/vms/vim-practice。不必事先创建此文件。原因见第二章第一节的说明。

```
root@HZ-CAS01-CVK01:~# vim /vms/vim-practice.txt
```

- 2、编辑文件

- a) 按i或者a进入插入模式(详细模式之间的转换见第一章的第三小节“vim的三种模式”)
- b) 按键键入要求显示的文本

- c) 按Esc以返回一般模式
- 3. 编辑已有文件，在文件底部编辑
  - a) 按o以在当前行的下方打开一个新行并立即从一般模式切换至插入模式
  - b) 按键键入要添加行的内容
  - c) 按Esc以返回一般模式
- 4. 在一般模式下按u键可撤销最近编辑
- 5. 按Ctrl+R恢复最近的撤销
- 6. 保存并退出

在一般模式中，输入:wq，然后按Enter即保存了编辑的文本并退出vim编辑器。