[**Linux文件编辑命令详细整理**](http://www.cnblogs.com/java-wp/p/7885732.html)

转载自-- [Linux文件编辑命令详细整理](http://blog.csdn.net/u013142781/article/details/50735470)

**一、vi编辑器有3种基本工作模式**

首先需要知道vi编辑器有3种基本工作模式，分别是：命令模式、文本输入模式、和末行模式。

第一：命令行模式：该模式是进入vi编辑器后的默认模式。任何时候，不管用户处于何种模式，按下ESC键即可进入命令模式。在该模式下，用户可以输入vi命令，用户管理自己的文档。此时从键盘上输入的任何字符都被当作编辑命令来解释。若输入的字符是合法的vi命令，则vi在接受用户命令之后完成相应的动作。但需要注意的是，所输入的命令并不回显在屏幕上。若输入的字符不是vi命令，vi会响铃报警。

第二：文本输入模式：在命令模式下输入命令i、附加命令a、打开命令o、修改命令c、取代命令r或替换命令s都可以进入文本输入模式。在该模式下，用户输入的任何字符都被vi当作文件内容保护起来，并将其显示在屏幕上。在文本输入过程中，若想回到命令模式下，按ESC键即可。

第三：末行模式：末行模式也称ex转义模式。在命令模式下，用户按“：”键即可进入末行模式下，此时vi会在显示窗口的最后一行（通常也是屏幕的最后一行）显示一个“：”作为末行模式的说明符，等待用户输入命令。多数文件管理命令都是在此模式下执行的（如把编辑缓冲区的内容写到文件中等）。末行命令在执行完后，vi自动回到命令模式。如果要从命令模式转换到编辑模式，可以键入a或者i。如果需要从文本模式返回，则按ESC即可。在命令模式下输入“:”即可切换到末行模式，然后输入命令。

**综上，一般我们使用命令打开文件的时候，是进入到命令模式。在命令模式下，可以切换到文本输入模式和末行模式，但是文本输入模式和末行模式之间是不可以直接相互切换了，因此文本输入模式切换到末行模式，需要先回到命令模式再切换，反之亦然。**

**此外，编辑文本可以在文本输入模式下，键盘输入对文编进行编辑，或者在命令模式下使用vi命令也是可以起到编辑效果的。**

**实例演示**

首先我们使用命令 vi filename 打开一个文件，这个时候进入到的是命令模式   
接下来我们按i，然后键盘随便输入写内容。   
然后按ESC重新进入到命令模式。   
在命令模式的情况下，我们按:，进入到了末行模式。   
我们输入wq!，然后回车，强行保存退出。   
下次我们再打开对应文件（可用less filename命令打开），即可看到内容已经更改。

**补充：**   
**（1）末行模式下：q! 【强制退出不保存】 q【退出不保存】 wq【退出并保存后面也可以加个！】**   
**（2）如果你不想保存直接退出有可以在命令模式下使用“ctrl+z”快捷键或者按住“shift”键，输入两个z即可退出。**

**更多命令**

**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 filename….filename ：打开多个文件，依次进行编辑

**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

**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

**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

**末行模式命令**   
：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。