**CentOS文件查看及编辑介绍**

时间:2015-03-25 09:55来源:lovesoo.org 作者:lovesoo.org [举报](http://www.centoscn.com/plus/erraddsave.php?aid=5014&title=CentOS%CE%C4%BC%FE%B2%E9%BF%B4%BC%B0%B1%E0%BC%AD%BD%E9%C9%DC) 点击:5688次

**2.1 cat 命令介绍**

cat 命令的原含义为连接(concatenate)， 用于连接多个文件内容并输出到标准输出流中（标准输出流默认为屏幕）。实际运用过程中，我们常使用它来显示文件内容。如果您熟悉MS-DOS 下的type 命令，相信不难掌握cat 命令。该命令的常用示例如下：

|  |  |  |
| --- | --- | --- |
| 1 | cat file1.txt 显示 file1.txt 文件的内容； | |
| 2 |  |

|  |  |  |
| --- | --- | --- |
| 3 | cat file1.txt file2.txt 显示 file1.txt 和file2.txt 文件的内容； | |
| 4 |  |

|  |  |
| --- | --- |
| 5 | cat -n file1.txt 显示 file1.txt 文件的内容同时显示行号； |

**2.2 vi 命令介绍**

vi 是”Visual Interface” 的简称，它在Linux 上的地位就仿佛Edit 程序在DOS上一样。它可以执行输出、删除、查找、替换、块操作等众多文本操作，而且用户可以根据自己的需要对其进行定制，这是其他编辑程序所没有的。

vi 不是一个排版程序，它不象Word 或WPS 那样可以对字体、格式、段落等其他属性进行编排，它只是一个文本编辑程序。

vi 没有菜单，只有命令，且命令繁多。限于篇幅，本文只介绍常用的命令。

Vi 有三种基本工作模式：命令行模式，文本输入模式和末行模式。

**命令行模式：**

任何时候，不管用户处于何种模式，只要按一下“ESC” 键，即可使vi 进入命令行模式；当在shell 环境下输入vi 命令启动vi 编辑器时，也是处于该模式下。

在该模式下 ，用户可以输入各种合法的vi 命令，用于管理自己的文档。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（但需注意的是，所输入的命令并不在屏幕上显示出来）。若输入的字符不是vi 的合法命令，vi 会响铃报警。

**文本输入模式：**

在命令模式下输入插入命令 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 当做文件内容保存起来，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。

**末行模式：**

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“:” 作为末行模式的提示符，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）末行命令执行完后，vi 自动回到命令模式。

若在末行模式下输入命令过程中改变了主意， 可按“ESC” 键或用退格键将输入的命令全部删除之后，再按一下退格键，即可使vi 回到命令模式下。

**2.2.1 vi 的进入与退出**

在 shell 模式下,键入vi 及需要编辑的文件名,即可进入vi. 例如:

|  |  |
| --- | --- |
| 1 | vi example.txt |

即可编辑 example.txt 文件.如果该文件存在,则编辑界面中会显示该文件的内容,并将光标定位在文件的第一行;如果文件不存在，则编辑界面中无任何内容。如果需要在进入vi 编辑界面后，将光标置于文件的第n 行，则在vi命令后面加上“+n” 参数即可。例如需要从example.txt 文件的第5 行开始显示，则使用如下命令：

|  |  |
| --- | --- |
| 1 | vi +5 example.txt |

退出 vi 时，需要在末行模式中输入退出命令“q”。 如果在文本输入模式下，首先按“ESC” 键进入命令模式，然后输入“:” 进入末行模式在末行模式下，可使用如下退出命令：

|  |  |  |
| --- | --- | --- |
| 1 | :q 直接退出。 如果在文本输入模式下修改了文档内容，则不能退出。 | |
| 2 |  |

|  |  |  |
| --- | --- | --- |
| 3 | :wq 保存后退出。 | |
| 4 | ：x 同“ wq”。 |

|  |  |
| --- | --- |
| 5 |  |
| 6 | :q! – 不保存内容， 强制退出。 | |

**2.2.2 vi 中显示行号**

在末行模式下，输入如下命令。

|  |  |
| --- | --- |
| 1 | set number |

可使 vi 在编辑界面中显示行号。

此外 ，在末行模式下，可使用如下“nu” 命令（number 的简写）来显示光标所在行的行号及该行的内容。

**2.2.3 光标移动操作**

全屏幕文本编辑器中， 光标的移动操作无疑是最经常使用的操作了。用户只有熟练地使用移动光标的这些命令，才能迅速准确地到达所期望的位置处进行编辑。

vi 中的光标移动既可以在命令模式下，也可以在文本输入模式下，但操作的方法不尽相同。

在文本输入模式下， 可直接使用键盘上的四个方向键移动光标；在命令模式下，有很多移动光标的方法。不但可以使用四个方向键来移动光标，还可以用h 、j、 k、 l 这四个键代替四个方向键来移动光标，这样可以避免由于不同机器上的不同键盘定义所带来的矛盾，而且使用熟练后可以手不离开字母键盘位置就能完成所有操作， 从而提高工作效率。

以下命令均在命令行模式下完成光标移动：

h 光标左移， 如果在按h 命令前输入数字n ，则光标左移n 个字符；

l 光标右移 ，如果在按l 命令前输入数字n， 则光标右移n 个字符；

j 光标上移， 如果在按j 命令前输入数字n， 则光标上移n 个字符；

k 光标下移， 如果在按k 命令前输入数字n ，则光标xia 移n 个字符；

0 （零） 光标移到行首；

$ 光标移到行尾；

H 光标移到屏幕上显示的第一行 （并不一定是文件头）；

L 光标移到屏幕上显示的最后一行 （并不一定是文件尾）；

M 光标移到屏幕的中间一行；

nG 光标移到第 n 行；

w或W 将光标右移至下一个单词的词首；

e 或E 如果光标起始位置处于单词内（即非单词尾处），则该命令将把光标移到本单词词尾；如果光标起始位置处于单词尾，则该命令将把光标移动到下一个单词的词尾。

b 或B 如果光标处于所在单词内（即非单词首），则该命令将把光标移至本单词词首；如果光标处于所在单词的词首，则该命令将把光标移到上一个单词的词首；

Ctrl+G 状态命令， 显示当前编辑文档的状态。包括正在编辑的文件名、是否修改过、当前行号、文件的行数以及光标之前的行占整个文件的百分比。

此外 ，也可以通过以下末行模式下的命令完成光标在整个文件中的移动操作：

:n 光标移到文件的第n 行；

:$ 光标移到文件的最后一行；

**2.2.4 文本插入操作**

在命令模式下 ,用户输入的任何字符都被vi 当作命令加以解释执行,如果用户要将输入的字符当作是文本内容时,则首先应将vi 的工作模式从命令模式切换到文本输入模式。切换的方式是使用下面的命令：

**1. 插入（Insert） 命令**

vi 提供了两个插入命令：i 和I。

（1） . i 命令插入文本从光标所在位置前开始， 并且插入过程中可以使用键删除错误的输入。此时vi 处于插入状态，屏幕最下行显示“–INSERT–” 插入字样。

（2） . I 命令 该命令是将光标移到当前行的行首， 然后在其前插入文本。

**2. 附加（append） 命令**

vi 提供了两个附加插入命令：a 和A。

（1） . a 命令该命令用于在光标当前所在位置之后追加新文本， 新输入的文本放在光标之后，在光标后的原文本将相应地向后移动，光标可在一行的任何位置。

（2） . A 命令 该命令与 a 命令不同的是，A 命令将把光标挪到所在行的行尾，从那里开始插入新文本，当输入A 命令后光，标自动移到该行的行尾。

a 和A 命令是把文本插入到行尾的唯一方法。

**3. 打开open 命令**

不论是 Insert 命令也好，还是append 命令也好，所插入的内容都是从当前行中的某个位置开始的。若我们希望在某行之前或某行之后插入一些新行，则应使用open 命令。

vi 提供了两个打开命令：o 和O。

（1） . o 命令该命令将在光标所在行的下面新开一行， 并将光标置于该行的行首，等待输入文本。要注意当使用删除字符时只能删除从插入模式开始的位置以后的字符，对于以前的字符不起作用。

（2） . O 命令和 o 命令相反，O 命令是在光标所在行的上面插入一行，并将光标置于该行的行首，等待输入文本。

**2.2.5 文本修改操作**

在命令模式下可以使用 vi 提供的各种有关命令对文本进行修改,包括对文本内容的删除、复制、取代和替换等。

**1. 文本删除/移动**

在编辑文本时 ，经常需要删除一些不需要的文本，我们可以用键将输错或不需要的文本删除，但此时有一个限制就是当删到行头之后，再想删上面那行的内容是不可能的。

在命令模式下， vi 提供了许多删除命令这些命令。大多是以d 开头的。常用的有：

（1） . 删除单个字符

x 删除光标处的字符。 若在x 之前加上一个数字n ，则删除从光标所在位置开始向右的n 个字符。

X 删除光标前面的那个字符，若在X之前加上一个数字n， 则删除从光标前面那个字符开始向左的n 个字符。

显然这两个命令是删除少量字符的快捷方法。

（2） . 删除多个字符

dd 删除光标所在的整行。 在dd 前可加上一个数字n ，表示删除当前行及其后n-1 行的内容。

D 或d$ 两命令功能一样， 都是删除从光标所在处开始到行尾的内容。

d0 删除从光标前一个字符开始到行首的内容。

dw 删除一个单词 。若光标处在某个词的中间，则从光标所在位置开始删至词尾。同dd 命令一样，可在dw之前加一个数字n 。表示删除n 个指定的单词。

如果用户不小心进行了误删除操作，也不要紧vi ，提供了恢复误操作的命令，并且可以将恢复的内容移动，放在文本的任何地方。恢复命令用“np”， 其中n 为需要恢复的次数。例如使用dd 命令删除了一行内容，然后使用“2p”命令，则被删除的内容会被重新插入两遍。

通过 dd 命令及p 命令的结合使用，可以很方便的实现文本行的移动操作。

**2. 文本复制命令**

yy 复制光标所在的整行。 在yy 前可加上一个数字n ，表示复制当前行及其后n-1 行的内容。

文本行复制后， 通过使用上面介绍的“p” 命令，可以将文本行粘贴到任何地方。

dd、 yy 及p 命令在vi 中的作用，类似于Windows 中图形编辑程序的“剪切”、“复制”及“粘贴”命令。

**3. 取消上一命令（Undo）**

取消上一命令 （Undo）， 也称复原命令，是非常有用的命令，它可以取消前一次的误操作或不合适的操作对文件造成的影响，使之回复到这种误操作或不合适操作被执行之前的状态。

取消上一命令有两种形式， 在命令模式下键入字符u 和U 它们的功能都是取消刚才输入的命令，恢复到原来的情况。小写u 和大写U 在具体细节上有所不同，二者的区别在于，大写U命令的功能是恢复到误操作命令前的情况，即如果插入命令后使用U 命令，就删除刚刚插入的内容；如果删除命令后使用U 命令，就相当于在光标处又插入刚刚删除的内容。这里把所有修改文本的命令都视为插入命令，也就是说U 命令只能取消前一步操作，如果用U命令撤消了前一步操作，当再按U 键时，并不是撤消再前一步的操作，而是撤消了刚才U命令执行的操作，也就是又恢复到第一次使用U 命令之前的状态，结果是什么都没做。而小写u 命令的功能是把当前行恢复成被编辑前的状态，而不管此行被编辑了多少次。

**4. 重复命令（Redo）**

重复命令也是一个非常常用的命令。 在文本编辑中经常会碰到需要机械地重复一些操作，这时就需要用到重复命令。它可以让用户方便地再执行一次前面刚完成的某个复杂的命令。

重复命令只能在命令模式下工作， 在该模式下按“.” 键既可。执行一个重复命令时，其结果是依赖于光标当前位置的。

**2.2.6 文本搜索与替换操作**

在进行文本编辑的时候,，您可以需要搜索或定位特定的单词或单词的一部分。vi 编辑器有向前或向后搜索指定模式的功能。

“/” 是vi 编辑器的模式搜索命令。键入“/” 后，屏幕的底部会出现一行，同时一个斜扛会出现在底行的行首上，且光标被放置在斜扛之后。此时，您可以在光标位置键入你需要搜索的模式。输入完成并按下 ENTER键后，编辑器将开始从光标所在的位置起向文件尾部搜索你键入的模式。如果输入的模式被找到，光标将停留在找到的模式上。在找到一个匹配的模式 后，如果您还需要搜索下一个匹配的模式，使用命令“n” 即可；如果需要找到上一个匹配的模式，则使用“N” 命令即可。

此外，“ ?” 命令也是模式搜索命令。与“/” 的区别在于/ 是从光标当  
前所在位置向文件尾部搜索，而“?” 命令则是由光标当前所在位置向文件头部搜索。

文本替换操作需要使用末行模式进行操作 即所有文本替换操作命令均以“:”开头。

文本替换命令的格式为：

:s/text1/text2 用于将光标所在段落搜索到的第一个 “text1” 替换为“text2”；

:s/text1/text2/g 用于将光标所在段落的所有 “text1 ”替换为“text2”；

:m,ns/text1/text2/g 用于将从 m行开始至n 行结束的所有搜索到的“text1”替换为text2。 其中可以使用$表示末行，即“1,$” 表示替换文档中的所有符合条件的字符。