

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

移动光标类命令

h : 光标左移一个字符

l : 光标右移一个字符

space: 光标右移一个字符

Backspace: 光标左移一个字符

k 或 **Ctrl+p**: 光标上移一行

j 或 **Ctrl+n** : 光标下移一行

Enter : 光标下移一行

w 或 **W** : 光标右移一个字至字首

b 或 **B** : 光标左移一个字至字首

e 或 **E** : 光标右移一个字至字尾

) : 光标移至句尾

(: 光标移至句首

}: 光标移至段落开头

{: 光标移至段落结尾

nG: 光标移至第 n 行首

n+: 光标下移 n 行

n-: 光标上移 n 行

n\$: 光标移至第 n 行尾

H : 光标移至屏幕顶行

M : 光标移至屏幕中间行

L : 光标移至屏幕最后行

0: (注意是数字零) 光标移至当前行首

\$: 光标移至当前行尾

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

最后行方式命令

: 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。

VI 常用技巧

VI 命令可以说是 Unix/Linux 世界里最常用的编辑文件的命令了, 但是因为它的命令集众多, 很多人都不习惯使用它, 其实您只需要掌握基本命令, 然后加以灵活运用, 就会发现它的优势, 并会逐渐喜欢使用这种方法。本文旨在介绍 VI 的一些最常用命令和高级应用技巧。

一、基本命令介绍

---- 1. 光标命令

k、j、h、l——上、下、左、右光标移动命令。虽然您可以在 Linux 中使用键盘右边的 4 个光标键, 但是记住这 4 个命令还是非常有用的。这 4 个键正是右手在键盘上放置的基本位置。

nG——跳转命令。n 为行数, 该命令立即使光标跳到指定行。

Ctrl+G——光标所在位置的行数和列数报告。

w、b——使光标向前或向后跳过一个单词。

---- 2. 编辑命令

i、a、r——在光标的前、后以及所在处插入字符命令(i=insert、a=append、r=replace)。

cw、dw——改变(置换)/删除光标所在处的单词的命令 (c=change、d=delete)。

x、d\$, dd——删除一个字符、删除光标所在处到行尾的所有字符以及删除整行的命令。

---- 3. 查找命令

---- /string、?string——从光标所在处向后或向前查找相应的字符串的命令。

---- 4. 拷贝复制命令

---- yy、p——拷贝一行到剪贴板或取出剪贴板中内容的命令。

二、常见问题及应用技巧

---- 1. 在一个新文件中读/etc/passwd 中的内容，取出用户名部分。

---- vi file

---- :r /etc/passwd 在打开的文件 file 中光标所在处读入/etc/passwd

---- :%s/:.*//g 删除/etc/passwd 中用户名后面的从冒号开始直到行尾的所有部分。

---- 您也可以在指定的行号后读入文件内容，例如使用命令":3r /etc/passwd"从新文件的第 3 行开始读入 /etc/passwd 的所有内容。

---- 我们还可以使用以下方法删掉文件中所有的空行及以#开始的注释行。

---- #cat squid.conf.default | grep -v ^\$ | grep -v ^#

---- 2. 在打开一个文件编辑后才知道登录的用户对该文件没有写的权限，不能存盘，需要将所做修改存入临时文件。

---- vi file

---- :w /tmp/1 保存所做的所有修改，也可以将其中的某一部分修改保存到临时文件，例如仅仅把第 20~59 行之间的内容存盘成文件/tmp/1，我们可以键入如下命令。

---- vi file

---- :20,59w /tmp/1

---- 3. 用 VI 编辑一个文件，但需要删除大段的内容。

---- 首先利用编辑命令"vi file"打开文件，然后将光标移到需要删除的行处按 Ctrl+G 显示行号，再到结尾处再按 Ctrl+G，显示文件结尾的行号。

---- :23,1045d 假定 2 次得到的行号为 23 和 1045，则把这期间的内容全删除，也可以在要删除的开始行和结束行中用 ma、mb 命令标记，然后利用":a,bd"命令删除。

---- 4. 在整个文件的各行或某几行的行首或行尾加一些字符串。

---- vi file

---- :3,\$s/^/some string / 在文件的第一行至最后一行的行首插入"some string"。

---- :%s/\$/some string/g 在整个文件每一行的行尾添加"some string"。

---- :%s/string1/string2/g 在整个文件中替换"string1"成"string2"。

---- :3,7s/string1/string2/ 仅替换文件中的第 3 行到第 7 行中的"string1"成"string2"。

---- 注意：其中 s 为 substitute，%表示所有行，g 表示 global。

---- 5. 同时编辑 2 个文件，拷贝一个文件中的文本并粘贴到另一个文件中。

---- vi file1 file2

---- yy 在文件 1 的光标处拷贝所在行

---- :n 切换到文件 2 (n=next)

---- p 在文件 2 的光标所在处粘贴所拷贝的行

---- :n 切换回文件 1

---- 6. 替换文件中的路径。

---- 使用命令":%s#/usr/bin#/bin#g"可以把文件中所有路径/usr/bin 换成/bin。也可以使用命令":%s//usr/bin//bin/g"实现，其中"\"是转义字符，表明其后的"/"字符是具有实际意义的字符，不是分隔符。