# VIM使用技巧

## vim中的三种模式

vim可以分为三种状态：一般模式、编辑模式和命令模式，各模式的功能区分如下：

一般模式：  
以vi打开一个文件就直接进入一般模式了(这是默认的模式)。在这个模式中， 你可以使用上下左右按键来移动光标，你可以使用删除字符或删除整行来处理文件内容， 也可以使用复制、粘贴来处理你的文件数据。

编辑模式：  
在一般模式中可以进行删除、复制、粘贴等的操作，但是却无法编辑文件的内容，只有当到你按下【i, I, o, O, a, A, r, R】等任何一个字母之后才会进入编辑模式。这时候屏幕的左下方会出现【INSERT或 REPLACE】的字样，此时才可以进行编辑。而如果要回到一般模式时， 则必须要按下【Esc】即可退出编辑模式。

命令行模式：  
输入【 : / ? 】三个中的任何一个，就可以将光标移动到最底下那一行。在这个模式中， 可以提供查找、读取、存盘、替换字符、离开vi、显示行号等的动作则是在此模式中完成的！

## 一般模式下的常用操作

移动光标

【h、j、k、l】分别控制光标左、下、上、右移一格，电脑上的↑↓←→下同样可实现此功能。

【ctrl+b】屏幕往“后”移动一页

【ctrl+f】屏幕往“前”移动一页

【G】移动到这个文件的最后一行

【nG】移动到这个文件的第n行(可配合:set nu)

【gg】移动到这个文件的第一行，相当于命令1G

【n<Enter>】光标向下移动n行

查找与替换

【/char】以当前光标为基准，向下寻找一个名称为char的字符串，同时将光标移动到此字符串上

【?char】以当前光标为基准，向上寻找一个名称为char的字符串，同时将光标移动到此字符串上

【n】代表重复前一个查找的动作

【N】 与n刚好相反，为【反向】进行前一个查找动作

【:n1,n2s/word1/word2/g】n1与n2为数字，在第n1与n2行之间查找word1 这个字符串，并将该字符串替换为word2

【:1,$s/word1/word2/g】从第一行到最后一行查找word1字符串，并将该字符串替换为word2  
【:1,$s/word1/word2/gc】从第一行到最后一行查找word1字符串，并将该字符串替换为word2 ，且在替换前提示用户确认是否进行替换

删除、复制和删除

【x】为向后删除一个字符 (相当于【del】键)  
【X】为向前删除一个字符(相当于【backspace】键)  
【nx】连续向后删除n个字符

【dd】删除光标所在行

【ndd】以当前光标为基准，向下删除n行

【d1G】删除光标所在行到第一行的所有数据  
【dG】删除光标所在到最后一行的所有数据

【d$】 删除光标所在处，到该行的最后一个字符  
【d0】 删除光标所在处，到该行的最前一个字符

【yy】复制光标所在的那一行

【nyy】以当前光标为基准，复制向下n行

【y1G】复制光标所在行到第一行的所有数据  
【yG】复制光标所在行到最后一行的所有数据

【y0】复制光标所在的那个字符到该行行首的所有数据（包含光标所在的字符）  
【y$】复制光标所在的那个字符到该行行尾的所有数据（包含光标所在的字符）

【p】将已复制的数据在光标下一行粘贴上  
【P】将已复制的数据在光标的上一行粘贴上

【u】恢复前一个操作

【.】重复上一个操作

## 一般模式进入编辑模式的常用操作

【i, I】进入编辑模式：  
i 为【从目前光标所在处插入】  
I 为【在目前所在行的第一个非空格符处开始插入】

【a, A】进入编辑模式：  
a 为【从目前光标所在的下一个字符处开始插入】  
A 为【从光标所在行的最后一个字符处开始插入】

【o, O】进入编辑模式：  
o 为【在目前光标所在的下一行处插入新的一行】  
O 为在目前光标所在处的上一行插入新的一行

【r, R】进入取代模式：  
r 只会取代光标所在的那一个字符一次  
R会一直取代光标所在的文字，直到按下 ESC 为止；

## 编辑模式退到一般模式

【Esc】 退出编辑模式，回到一般模式

## 一般模式切换到命令模式的常用操作

【:w】保存编辑的内容

【:w!】强制写入该文件，但跟你对改文件的权限有关

【:q】离开vim

【:q!】不想保存修改，强制离开

【:wq】保存后离开

【:wq!】保存后，强制离开

【:x】保存后离开

对于笔者，经常使用的是【:q!】【:wq!】这两个命令。

【:w filename】将编辑的数据保存成另一个文件（类似另存）  
【:r filename】在编辑的数据中，读入另一个文件的数据。即将【filename】这个文件的内容加到光标所在行后面。

【:n1,n2 w filename】 将n1到n2的内容保存成filename这个文件。

【:set nu】显示行号（经常使用）  
【:set nonu】与 set nu 相反，为取消行（经常使用）

上述说的基本上是vi的使用技巧，下面说说vim中个人感觉比较高级用法。

## vim中的高级用法

块选择：

【v】字符选择，会将经过的的地方反白选择

【V】行选择，会将光标经过的行反白选择

【Ctrl+v】块选择，可以用长方形的方式选择资料。

【y】将反白的地方复制

【d】将反白的地方删除

多文件编辑

vim可以进行多文件打开，进行切换，多文件之间可以进行复制、粘贴

【vim file1 file2】打开多个文件

【:n】编辑下一个文件

【:N】编辑上一个文件

【:files】列出目前这个vim编辑的所有文件

多窗口功能

在打开一个文件的基础上，打开另一个文件

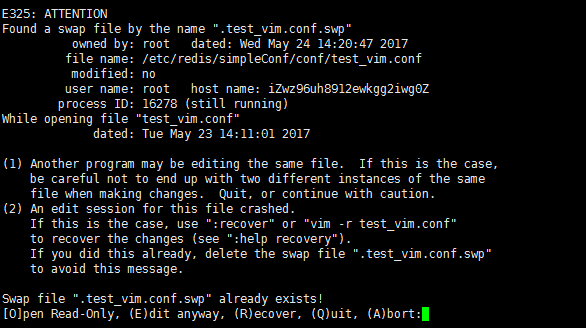
【:sp filename】开启一个新窗口，如果有加filename，表示在新窗口开启一个新文件，否则表示两个窗口为同一个文件内容

【Ctrl+w+j】【Ctrl+w+↓】按键的按法是：先按Ctrl不放，再按下w后放开所有的按键，然后在按下j（或↓），则光标可移动到下方的窗口。

【Ctrl+w+k】【Ctrl+w+↑】同上，光标可移动到上方的窗口。

## 问题总结

1. 有时通过vim打开文件时，会出现下面的场景：



这个是怎么导致的呢？

这个是vim的一个恢复功能，就是说当你的系统因为某些原因而导致类似一些宕机的情况时，还可以利用这个恢复功能将之前未保存的数据找回来。

当我们使用vim进行编辑时，vim会在与被编辑的文件所在的目录下，创建一个.filename.swp文件，当系统出现异常，或者一些非法操作，此文件会依然保存，此时这个文件就可以发挥一个救援的功能。

问题重现：

1. vim test\_vim.conf
2. Ctrl+z,将vim放到后台运行
3. ps aux |grep vi kill -9 n 将vim进程杀掉

此时在当前目录下通过ls命令可以看到.test\_vim.conf.swp文件，如果看不到，vim .双击tab键即可看到此文件。

上述图片说明了出现这些提示的原因：

1. 可能有其他人或程序同时在编辑这个文件。
2. 在vim编辑时，可能因为某些不知名原因导致vim中断。

图片中的最下方出现了5个命令项，功能如下：

[O]pen Read-Only：打开此文件成为只读档，可以用在你只想查阅该文件内容并不想进行编辑时。

(E)dit anyway：用正常的方式打开你要编辑的那个文件， 并不会载入暂存盘的内容。如果说两个人都在编辑这个文件的话，很容易出现互相改变对方的文件等问题。

(R)ecover：加载暂存盘的内容，用在你要救回之前未保存的工作。不过当你救回来并且储存离开vim后，还是要手动自行删除那个暂存档。

(Q)uit：按下q 就离开vim，不会进行任何动作回到命令提示界面。

1. bort：忽略编辑行为，退到命令界面，类似Q。

应该还有一个：

(D)elete it:开启文件前，先将这个暂存盘删掉。

1. 有这种情况，当以vim软件查找一个文件里的某个字符时，这个字符串会被反白，当下次我们再以vim编辑这个文件时，字符串的反白情况依然存在，另外光标会停留在我们上次编辑的位置，这个是怎么回事呢？

原来vim会主动将我们曾经做过的操作记录下来，记录的文件是：~/.viminfo，不想用这个功能，可以直接把这个文件删除掉，但下次使用vim时，会再次生成。