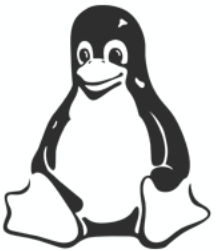


Linux系统管理

南通师范高等专科学校 朱亚林

本章导言

能否高效、便捷地编辑各类文件是衡量一个操作系统是否出色的重要指标。在Linux的各类桌面版中（GNOME、KDE、XFCE等），各类图形界面的编辑器百花齐放，例如办公套件WPS、Libreoffice等、编程软件geany、vscode等，但回到命令行下，就不得不提一款功能强大的编辑利器——VI/VIM。

本章将就VI/VIM的高效使用进行介绍。

第6章

编辑利器：VI/VIM

6.1 VI/VIM简介

- Vi是Unix及Linux系统下标准的编辑器，由美国加州大学伯克利分校的Bill Joy开发问世。
- 荷兰的Bram Moolenaar（布莱姆·米勒）在上个世纪80年代末购入一台计算机，比较遗憾的是那台计算机上没有他最常用的Vi编辑器。于是他复制了Vi的一个开源分支Stevie，开发了Vim 1.0版本，那个时候的 Vim 是Vi IMitation（模拟）的简称。后来由于加入的特性越来越多，也越来越受到人们的喜爱和欢迎，Bram Moolenaar也顺势将 Vim的名字全称改为了Vi IMproved，意为Vi的增强版本。Bram Moolenaar在开发Vim的同时，还一直在资助乌干达改善儿童教育和医疗事业。

6.2.1 Vim的安装

- Debian中Vim的安装在第二章中已经做过介绍。通过在命令行中输入：

```
sudo apt install vim
```

6.2.2 Vim的启动

- Vim的启动也是通过命令行来实现的，一般格式如下：

```
vim [文件名]
```

注意，此处的vim是小写字母。

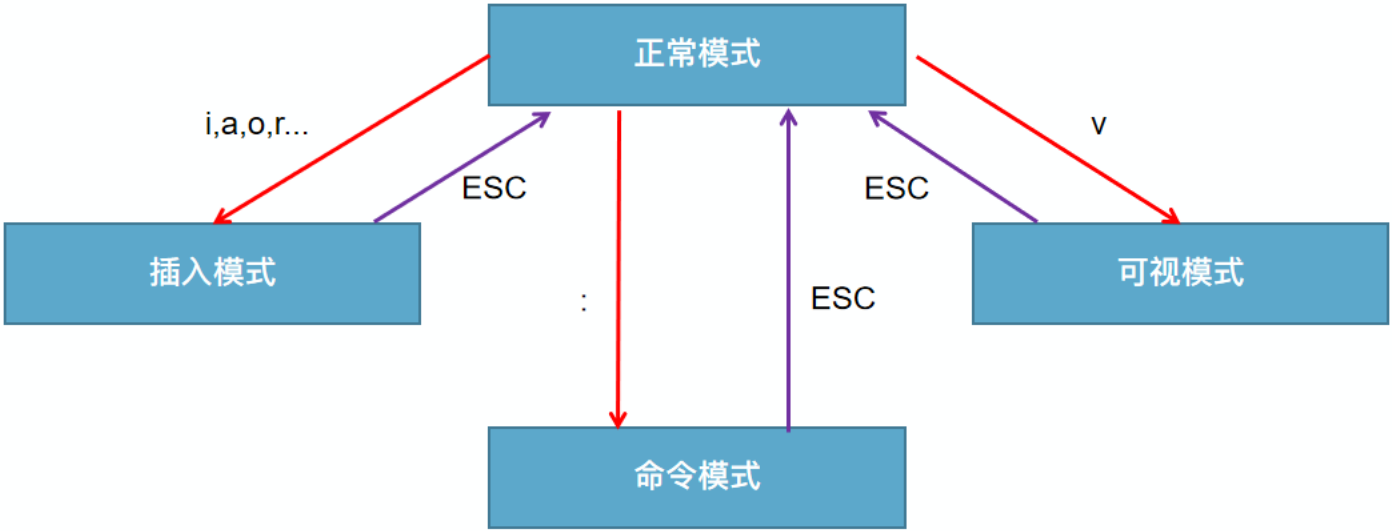
- 如果只是输入vim，则直接启动编辑器；
- 如果在Vim后输入文件名，该文件已经在当前路径中存在，则为编辑该文件；
- 如果在Vim后输入文件名，该文件在当前路径中不存在，则为新建文件。

6.2.3 Vim的四种常用模式

- Vim不同于常见的编辑器，并不是打开就可以直接操作。在Vim中有四种常用模式，分别是：正常模式、插入模式、命令模式和可视模式。

- 正常模式(Normal-mode)：当编辑器启动后默认新状态下就是进入了正常模式。此时可以进行光标移动，对文本进行复制、粘贴、删除等基本操作；
- 插入模式(Insert-mode)：当按下了“i、I、o、O、a、A、r、R”键后，编辑器进入到插入模式，这个模式就和Windows下记事本有些相似，可以进行文本的输入，以及相应的编辑。
- 命令模式 (Command-mode)：当按下“ : / ?”等键后，光标会在编辑器最下方进行闪烁，此时就是进入到了命令模式，输入相关的指令执行诸如：保存、退出、查找、替换等等一系列操作。
- 可视模式 (Visual-mode)：在正常模式中按下v, V, <Ctrl>+v，可以进入可视模式。可视模式中的操作类似于使用鼠标进行操作，如果支持鼠标的话可以使用鼠标选择文本。

6.2.3 Vim的四种常用模式



6.2.4 Vim的退出

VIM的退出命令是 :wq

- “Esc”触发正常模式,
- “:”触发命令模式,
- wq是write和quit的缩写，意为保存退出，其后的文件名为保存路径及文件名；如果当前编辑的文件已经存在，则直接输入“:wq”即可。

6.3 Vim常用操作

- 在Vim插入模式下，它就是一个文件编辑器。
- 正常模式和命令模式是Vim区别于其他编辑器的地方，是它的优长所在，要依靠大量的快捷键和指令来完成。

6.3.1 插入模式

插入（insert）

- 按下i键，从光标所在位置前面插入文本，光标后的数据随着新增文本向后顺次移动。
- 按下I键，从光标所在行的第一个非空格字符前面开始插入文本，其后文本顺次后移。

新增（append）

- 按下a键，从光标所在位置后面插入文本，光标后的数据随着新增文本向后顺次移动。
- 按下A键，从光标所在行的行尾插入文本。

开始（open）

- 按下o键，在光标所在行下方新增一行，并进入输入模式。
- 按下O键，在光标所在行上方新增一行，并进入输入模式。

Esc

- 按下Esc键，退出插入模式

6.3.1 插入模式

案例

- 使用vim编辑ex-1.cpp，并输入如下图中所示内容

```
#include <iostream>
using namespace std;
int main()
{
    string s="you are using VIM";
    cout << s << endl;
}
                                                    6,22          All
```

6.3.2 光标的控制

以行列为单位移动光标

在正常模式和插入模式下，都可以通过上下左右方向键来移动光标。同时Vim也内置了几个简单的光标移动键，方便在正常模式下进行光标控制。具体如下。

- h：向左移动一列
- l：向右移动一列
- j：向下移动一行
- k：向上移动一行

6.3.2 光标的控制

案例：在Vim中快速移动光标

（1）使用Vim打开LinuxStudy/chapter-6中的ex-1.cpp

（2）将光标快速定位到第5行

（3）将光标快速定位到当前行第10列

（4）将光标上移两行

（5）将光标向前移3个位置

6.3.2 光标的控制

以单词为单位移动光标

在Vim中除了可以以行列为单位进行光标移动外，还可以以单词为单位进行光标前后移动。具体如下。

- w：将光标移动到下一个单词的开头
- b：将光标移动到前一个单词的开头
- e：将光标移动到下个单词的末尾

6.3.2 光标的控制

以行号定位光标

Vim支持快速定位到指定行。为了让光标定位更加直观，先将Vim显示行号的功能开启。具体方法是在正常模式下输入“:”进入命令模式，然后输入“set number”回车即可。

6.3.2 光标的控制

- 快速移动到指定行的方法如下表所示

命令	作用
–nG	将光标移动到第n行
G	将光标移动到最后一行
gg	将光标移动到第一行
:n	将光标移动到第n行
:$	将光标移动到最后一行
:set number	开启Vim行号显示
:set nonumber	关闭Vim行号显示

6.3.2 光标的控制

- 行内光标的快速定位

命令	作用
$	将光标移动到当前行的行尾
^或0	将光标移动到当前行的行首
n|	将光标移动到当前行的第n个字符上
fm	将光标移动到当前行下一个字符m上

6.3.2 光标的控制

- 其他定位方式

命令	作用
H	将光标移动到屏幕的最上面1行
M	将光标移动到屏幕的中间行
L	将光标移动到屏幕的最下面1行
Ctrl-D	下移半屏
Ctrl-F	前进一屏
Ctrl-U	上移半屏
Ctrl-B	后退一屏

6.3.3 Vim的编辑功能

1. 文本的插入

- 本节“切换到插入模式”中重点讲述了文本的插入方式，这里再来重温一下。
- 步骤：

（1）使用Vim打开ex-1.cpp

（2）设置Vim显示行号

（3）在第5行之后插入如下内容

string m = “Yes. It’s amazing. I Loved it”;

（4）保存以上内容

6.3.3 Vim的编辑功能

2. 文本的删除

Vim中删除文本可以切换到插入模式，使用delete或者backspace键，也可以使用如下表所示的各类命令按键。

命令	作用
x	删除光标所处位置的字符
dw	删除光标所处位置的字（单词）
D	删除光标向后直到所在行末尾的内容
dd	删除光标所在行
ndd	删除光标所在行向后的n行内容
:<m,n>d	删除指定范围内容。<m,n>是指定一个范围，如3,5

6.3.3 Vim的编辑功能

- 案例：

（1）对于刚刚插入的文本，通过观察可以发现，Loved这个单词在语法上存在一定问题，需要将最后的d删除掉。

（2）将第6行中的Yes单词删除

（3）将第5行删除。

6.3.3 Vim的编辑功能

✓ 3. 文本的替换

文本的替换可以使用以下命令按键来完成。

命令	作用
r	替换光标所在位置的字符
R	逐字符替换光标之后的文本，完成替换后按Esc键
cw	替换光标所在位置的字（单词）
cc	替换光标所在行的内容

6.3.3 Vim的编辑功能

- 案例：编辑ex-1.cpp，替换指定内容

（1）将第5行的“s”替换成“m”

（2）将第5行的“it”替换成“Vim”

（3）将第6行的内容替换成printf(“%s”,m);

6.3.3 Vim的编辑功能

4. 文本的复制、粘贴与移动

- Vim中默认文本的复制与粘贴都是以行为单位的。先将指定区域的内容复制到缓冲区，然后再粘贴到指定位置。

命令	作用
:<m,n>y	复制从m行向后的n行到粘贴缓冲区
Y	复制当前行到粘贴缓冲区
p或:pu	在当前行之后插入粘贴缓冲区
:*n*pu	在n行之后插入粘贴缓冲区
yf*m*	复制光标当前位置到第一个m字符到粘贴缓冲区
yw	复制光标所在位置的字（单词）到粘贴缓冲区
yl	复制光标所在位置的字符到粘贴缓冲区

6.3.3 Vim的编辑功能

案例：编辑ex-1.cpp，进行复制移动操作

步骤：

（1）使用Vim打开ex-1.cpp

（2）使用两种方法，将3-7行内容复制到第7行以后

（3）将1-2行移动到第7行以后

（4）使用两种方法，删除1-5行内容

6.3.3 Vim的编辑功能

5. 文本的查找和替换

- Vim中激活文本查找功能的按键有两个，分别是“/”和“?”（不含引号）。具体如下表所示。

命令	作用
/string	从光标位置顺向查找字符串string
?string	从光标位置反向查找字符串string
n	在当前方向上继续查找
N	在反方向上继续查找

6.3.3 Vim的编辑功能

5. 文本的查找和替换

- 对于字符串的替换，具体操作如下表所示

命令	作用
:<m,n>s/s1/s2	在m到n行内，将每行中第1次出现的s1替换成s2
:<m,n>s/s1/s2/g	在m到n行内，将每行中所有的s1替换成s2

说明，如果替换范围是全篇，则可以全用1,$来表示。

6.3.3 Vim的编辑功能

- 案例：打开 LoveYourLife.txt，进行查找替换操作

（1）从http://t.edutest.fun/c下载文件，并更名为LoveYourLife.txt

（2）查找全文中的house单词

（3）将全文中的Your改成My

6.3.3 Vim的编辑功能

- 6. Vim其他常用功能

命令	作用
u	撤销上一步
U	撤销对当前行的多个操作
.	重做上一步
CTRL-L	重绘屏幕
J	连接当前行和下一行
‘’	连续两次单引号，快速切换当前位置和文档起始位置
:q!	不保存强制退出
:q	退出
:w	保存
:wq	保存后退出
:x	保存后退出