项目3 文本编辑器 vim

【本章学习目标】

- (1) 了解 vim 和 vi 的区别。
- (2) 掌握 vim 的 3 种工作模式。
- (3) 掌握一般模式下的快捷键。
- (4) 掌握命令行模式下的快捷键。
- (5) 熟悉 Linux 的文件救援和高级功能。

vi编辑器(Visual Editor)通常简称为vi,是一种命令行界面下的文本编辑器。在早期的UNIX操作系统中都是使用vi作为系统默认的编辑器的。vim(Vi IMproved)就是vi的升级版,vim和vi最大的区别在于,在编辑一个文本的时候,vi不会显示颜色,而vim会显示颜色。此外,vim还能够进行Shell脚本、C语言源程序、Java等程序的编辑,使用vim能帮助程序员更容易找出源程序中的语法错误,因此可以将vim视为一种程序编辑器。

在 CentOS 8.1 系统中,已经默认安装有 vim 文本编辑器,所以不需要安装。如果目前 Linux 系统中没有 vim 命令,可在计算机连网后使用命令 yum install -y vim-enhanced 自行 进行安装。

3.1 vim 的工作模式

vim 是怎么工作的呢?首先来了解一下 vim 的工作模式。vim 的工作模式有一般模式 (也称指令模式)、编辑模式和命令行模式 3 种。

3.1.1 vim 的 3 种工作模式

1. 一般模式

一般模式有时也称为指令模式。当使用命令 vim filename 打开一个文件时,一进入该 文件,就是一般模式了。在这种模式下,可以上、下、左、右移动光标,直接删除某个字符或删 除某些行,复制一行或者多行,进行粘贴,以及查找并替换字符或字符串。因此,一般模式下 的功能键主要有 3 类:移动光标类,删除、复制和粘贴类,查找替换类。

2. 编辑模式

在一般模式下,是不可以修改某一个字符的,要想修改,只能进入编辑模式。要想从一 般模式切换到编辑模式,只需要按i、I、a、A、o、O、r和R这8个键中的任意一个即可。这8 个键的意义是不一样的,后面的表格中会有详细的解释。这8个键中必须记住的是i键。

当进入编辑模式后,屏幕的最后一行会出现"一插入一"或者"一替换一"的字样。 如果想从编辑模式返回到一般模式,按 Esc 键即可。

在编辑模式下,主要是编辑文档内容,进行文本的插入或者替换等修改操作。

3. 命令行模式

在一般模式下,当输入":",就进入了命令行模式。在命令行模式下,可以进行的操作有保存文件、退出 vim、读入外部文件、设置行号和取消行号等。

通常情况下,可以将这3种模式想象成一幅图。图 3-1 就是 vim 3 种模式之间的转换 关系图。认真看上面的图标,会发现一般模式可以与编辑模式和命令行模式相互切换,但是 编辑模式与命令行模式之间是不可以互相切换的,这一点非常重要。



图 3-1 vim 3 种模式之间的转换关系图

3.1.2 运行 vim

在终端中输入 vim filename 命令,然后按 Enter 键,就进入 vim 的一般模式了。如果当前目录中存在这个文件,则打开该文件;如果不存在这个文件,则新建文件,因此,该命令无论什么时候都不会报错。

上面已经讲解了 vim 的 3 种工作模式以及它们之间的转换关系。下面举个简单的例子 让大家加深印象,要求使用 vim 命令创建一个文本文件 test.txt,并在其中写入一段文字,然 后进行保存,最后退出 vim。具体步骤如下。

(1) 打开终端,在命令行提示符后输入命令"vim test.txt",然后按 Enter 键,这样就进 入了 vim 的工作界面。此时,光标在第一行的行首闪烁,对话框的最下面显示出文件名 "text.txt"。"[]"中的"新文件"3 个字表示这是一个新的文件。现在处于 vim 的一般模式, 如图 3-2 所示。

		r	oot@sjh:~			×
文件(F)	编辑(E)	查看(V)	搜索(S)	终端(T)	帮助(H)	
~						
~						
- · ·						
÷ .						
-						
~						
						2.2-
				0 0 1		

(2) 按i键,屏幕的最后一行会出现"--插入--"字样,此时处于 vim 的编辑模式,随便输

入一些字符,如图 3-3 所示。

(3) 输入完毕后,按 Esc 键,屏幕下方的"-- 插入 --"消失,返回到一般模式下。

(4) 输入":",进入命令行模式,接着输入"w",此时":w"会显示在窗口的最后一行,如 图 3-4 所示,然后按 Enter 键,文件内容将会被写入,也即文件内容已保存。此时,窗口的最 后一行会显示"已写入"并显示出文件的总行数和总字符数,当前文件的总行数为 11 行,总 字符数为 504,写入后的文件如图 3-5 所示。



图 3-3 vim 的编辑模式

图 3-4 输入"w"将会写入文件

root@sjh:~	×
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)	
his is a test file. And this is the first time to use "vim". It's easy to use it. I like using "vim",do you like it? ~ ~	
~ "test.txt" [新] 11L, 504C 已写入 1,1	顶端

图 3-5 写入后的文件

(5) 输入":",再输入"q",此时":q"会显示在窗口的最后一行,如图 3-6 所示,按 Enter 键,退出 vim。

	root@sjh:~	×
文件(F) 编辑(E) 查看	(V) 搜索(S) 终端(T) 帮助(H)	
This is a test fil And this is the fi It's easy to use i I like using "vim" ~ ~ ~ ~ :q	e. rst time to use "vim". t. ,do you like it?	

图 3-6 输入"q"将会退出 vim

至此,文件 test.txt 已经生成。使用 11 命令进行查看,如图 3-7 所示。

			root@s	sjh:~		×
文件(F) 编辑(E)	查看(V)	搜索(S)	终端((T)	帮助(H)	
[root@sjh ~]# 总用量 8	11					
drwxr-xr-x. 2	root root	t 6	3月	29	16:55	公共
drwxr-xr-x. 2	root root	t 6	3月	29	16:55	模板
drwxr-xr-x. 2	root root	t 6	3月	29	16:55	视 频
drwxr-xr-x. 2	root root	t 6	3月	29	16:55	图片
drwxr-xr-x. 2	root root	t 6	3月	29	16:55	文档
drwxr-xr-x. 2	root root	t 6	3月	29	16:55	下载
drwxr-xr-x. 2	root root	t 6	3月	29	16:55	音乐
drwxr-xr-x. 2	root root	t 6	3月	29	16:55	桌面
-rw 1	root root	t 1750	3月	6	21:58	anaconda-ks.cfg
-rw-rr 1	root root	t 0	3月	6	22:21	initial-setup-ks.cfg
-rw-rr 1	root root	t 504	4月	1	23:39	test.txt
[root@sjh ~]#			-			

图 3-7 查看 test.txt 文件信息

3.2 vim 常见命令

通过 3.1 节中的例子,相信大家都已经学会怎么使用 vim 来进行简单的文本编辑了。 但其实 vim 是全键盘式操作的编辑器,所以在各个模式下都有很多的功能键,只有掌握好 了这些功能键,才能成为真正的 vim 高手。下面将对 3 种模式下的功能键进行详细的讲解。

3.2.1 一般模式下的功能键

在一般模式下可以使用的功能键最多,大致可以分为3类,第一类是移动光标类,第二 类是删除、复制和粘贴类,第三类是查找和替换类。

1. 移动光标类

一般模式下的移动光标类快捷键如表 3-1 所示。

按键名称	按键的效果
h或者←键	光标向左移动一个字符
j或者↓键	光标向下移动一个字符
k或者≮键	光标向上移动一个字符
]或者→键	光标向右移动一个字符
Ctrl+f 键或者 PgDn 键	屏幕向文件尾部移动一页
Ctrl+b 键或者 PgUp 键	屏幕向文件头部移动一页
n 空格(n 是数字)	按下数字 n 然后按空格,则光标向右移动 n 个字符,如果该行字符数小于 n,则光标继续从下一行开始向右移动,一直到 n
0(数字 0)或者 Home	移动到本行行首
\$ 或者 End	移动到本行行尾
Н	光标移动到当前屏幕的最顶行

表 3-1 一般模式下的移动光标类快捷键

按键名称	按键的效果
М	光标移动到当前屏幕的中央那一行
L	光标移动到当前屏幕的最底行
G	光标移动到文件的最后一行
nG(n 是数字)	光标移动到文件的第 n 行
gg	光标移动到文件的第一行
n 回车(n 是数字)	光标向下移动 n 行

使用键盘上的 ↑、√、←、→键,可以将光标移动一个字符,也可以使用键盘上的 h、j、k、l 来实现同样的功能。如果要移动多个字符,可以按 n+方向键,这里,n 代表一个数字。例 如,如果想将光标向左移动 8 个字符,先按 8 键,再按 h 键,就可以了。或者也可以先按 8, 再按←键,光标也会向左移动 8 个字符。空格代表向右移动一个字符,如果想将光标向右移 动 20 个字符,可以按 201(小写的英文),20→,或者 20 空格,都可以。

一个大写的 G 可以将光标移动到文件的最后一行。nG 将光标移动到文件的第 n 行, 例如,想移动光标到第 101 行,按 101G 就可以。要把光标移动到文件的第 1 行,可以使用 1G 或者 gg。

*n*Enter,表示将光标从当前行开始向下移动*n*行。如果当前光标位于第 10 行,按 5Enter 后,光标将会移动到第 15 行。

2. 删除、复制和粘贴类

除了移动光标,还经常进行删除、复制和粘贴的操作。一般模式下的删除、复制和粘贴 类快捷键如表 3-2 所示。

按键名称	按键的效果
x,X	x表示向后删除一个字符,X表示向前删除一个字符
nx(n 是数字)	向后删除 n 个字符
dd	删除光标所在的那一行
ndd(n 是数字)	删除光标所在的向下 n 行
d1G	删除光标所在行到第1行的所有数据
dG	删除光标所在行到末行的所有数据
уу	复制光标所在的那一行
пуу	复制从光标所在行开始的向下 n 行
p,P	p将复制的数据从光标下一行粘贴,P则从光标上一行粘贴
y1G	复制光标所在行到第1行的所有数据
уG	复制光标所在行到末行的所有数据

表 3-2 一般模式下的删除、复制和粘贴类快捷键

续表

• 82 •

续表

按键名称	按键的效果
J	将光标所在行与下一行的数据结合成一行
u	还原过去的操作
Ctrl+r	重做上一个操作
	重复前一个操作

如果想将光标之后的 15 个字符删除,按 15x 即可。如果要删除光标之前的 20 个字符, 按 20X 即可。

按 dd 将删除光标所在的那一行,按 ndd 将删除包括光标所在行开始的向下 n 行。如果光标位于第 11 行,想将第 11~22 行之间(包括第 11 行和第 22 行)的文本删除,只需按下 12dd 即可。

3. 查找和替换类

最后一类为查找和替换类。一般模式下的查找和替换类快捷键如表 3-3 所示。

表 3-3 一般模式下的查找和替换类快捷键

按键名称	按键的效果
/keyword	向光标之后查找名为 keyword 的字符串,当找到第一个 keyword 后,该单词 高亮显示,按n键继续查找下一个,按N键,反方向查找下一个
? keyword	向光标之前查找名为 keyword 的字符串,当找到第一个 keyword 后,该单词 高亮显示,按 n 键继续查找下一个,按 N 键,反方向查找下一个
:n1,n2s/word1/word2/g	在 nl 和 n2 行之间查找 word1 字符串并替换为 word2
:1, \$ s/word1/word2/g	从第一行到最末行,查找 word1 并替换成 word2
:1,\$ s/word1/word2/gc	在第一行和最末行之间查找 word1, 替换为 word2 之前需要用户确认

3.2.2 从一般模式进入编辑模式

从一般模式进入编辑模式,需要按 i、I、a、A、o、O、r、R 这 8 个键中的任意一个,这 8 个键的意义是不一样的。其中,按 i、I、a、A、o、O 时,窗口最下方将会出现"-- 插入 --"字样,按 R 键时,窗口最下方将会出现"-- 替换 --"字样。

从一般模式进入编辑模式的8个键如表3-4所示。

按键名称	按键的效果
i	在光标前插入字符
Ι	在光标所在行的行首插入字符
a	在光标后插入字符
А	在光标所在行的行末插入字符

表 3-4 从一般模式进入编辑模式的 8 个键

• 83 •

续表

按键名称	按键的效果
0	在光标所在行的下面插入新的一行
0	在光标所在行的上面插入新的一行
r	替换光标所在的字符,只替换一次
R	一直替换光标所在的字符,直到按 Esc 键

3.2.3 命令行模式下的功能键

在一般模式下,输入":"将进入命令行模式。命令行模式下支持的功能有保存文件、退 出 vim、读入外部文件、设置行号等。命令行模式下的功能键如表 3-5 所示。

按键名称	按键的效果
: W	保存文件内容
: w!	如果文件属性为只读时,强制保存
:q	退出 vim
:q!	强制退出 vim,不管编辑还是未编辑都不保存内容直接退出
:wq	保存文件内容之后立即退出
:e!	将文档还原成最原始状态
ZZ	等价于:wq
:w[filename]	将文档另存为 filename
r [filename]	在光标所在行的下面读入 filename 文档的内容
:set nu	在文件中每行的行首设置行号
:set nonu	取消已经设置的行号
$:n_1, n_2 \text{ w [filename]}$	将 $n_1 \sim n_2$ 行的内容另存为 filename 文件中
:! command	暂时离开 vim,执行某个 Linux 命令,例如: :! ls /home 暂时列出/home 下的文件,然后会提示按 Enter 键返回 vim

表 3-5 命令行模式下的功能键

【想一想】 在 vim 中,将打开的文件另存之后并没有退出 vim,接着继续编辑文件时, 必须知道继续编辑的是原来的文件还是另存后的文件。

3.3 vim 的其他事项

本节将介绍文件救援、多窗口编辑、多文件编辑和块选择等 vim 的其他事项。

3.3.1 文件救援

在使用 vim 的过程中可能会出现文件还没来得及保存,计算机突然断电或者不小心关

• 84 •

闭了终端的情况,这时候要挽救没有保存的文件,就要使用 vim 的救援功能了。

在使用 vim 编辑文件时, vim 会在被编辑的文件的目录下,再建立一个名为.filename.swp 的文件。如果系统因为某些原因突然断线了,导致编辑的文件还没有及时保存,这个时候. filename.swp 就能够发挥救援的功能了。当再次使用命令 vim filename 打开文件时,将会 弹出发现交换文件.filename.swp 的警告信息。

本例在使用 vim test.txt 命令编辑文件后没有保存就退出了终端,再次使用 vim test.txt 命 令时就弹出了如图 3-8 所示的警告信息。

root@sih:~ × 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H) E325: 注意 发现交换文件 ".test.txt.swp 所有者: root 日期: Thu Apr 2 00:22:47 2020 文件名: ~root/test.txt 修改过:是 用户名: root 主机名: sjh.com 进程 ID: 5372 正在打开文件 "test.txt" 日期: Wed Apr 1 23:39:25 2020 (1) Another program may be editing the same file. If this is the case, be careful not to end up with two different instances of the same file when making changes. Quit, or continue with caution. (2) An edit session for this file crashed. 如果是这样,请用 ":recover" 或 "vim -r test.txt" wu未定这件, 項用 ":recover" 或 "vim -r test.txt" 恢复修改的内容 (请见 ":help recovery")。 如果你已经进行了恢复,请删除交换文件 ".test.txt.swp" 以避免再看到此消息。 交换文件 ".test.txt.swp" 已存在! 以只读方式打开([0]), 直接编辑((E)), 恢复((R)), 删除交换文件((D)), 退出((Q)), 中止((A)): ■

图 3-8 发现交换文件的警告信息

在警告信息的画面中,有6个可用选项。

(1) [O]pen Read-Only: 以只读方式打开。

(2)(E)dit anyway:直接编辑。

(3) (R) ecover:恢复,即加载暂存盘的内容,用来挽救之前未保存的工作。

(4) (D) elete it: 删除文件,如果确定那个暂存文件是无用的,那么可以先将这个暂存 文件删除。有时候如果不确定这个暂存文件是怎么来的,也可以删除它。

(5)(Q)uit:退出,按q键就可离开 vim,不会进行任何动作,返回到命令行提示符。

(6)(A)bort:终止,与 quit 差不多,也会返回到命令行提示符。在这里,先按 R 键,再按 Enter 键,test.txt 文件就显示出来了,恢复之后的文件如图 3-9 所示。

需要注意的是,除非选择(D)elete it 删除了该交换文件,否则当离开 vim 后,还需要在终端中执行命令

rm .test.txt.swp

自行删除该交换文件,不然,以后每次使用命令

vim test.txt

都会出现同样的警告信息。

			roc	root@sjh:~					
文件(F)	编辑(E)	查看(V)	搜索(S)	终端(T)	帮助(H)				
This is And thi It's ea I like There a ~	a test s is th sy to u using " re thre	file. e first se it. vim",do e modes	time t you li in vim	o use "\ ke it?	vim".				
÷~					5,29	全部			

图 3-9 恢复之后的文件



3.3.2 多窗口编辑

当使用 vim 编辑文件时,还可以将不同的文件同时显示在屏幕的不同窗口中,也可以 将同一个文件同时显示在屏幕的不同窗口中,实现文件的多窗口编辑,也即常说的分屏 功能。

1. 多个文件分屏编辑

命令格式如下:

vim -On [FILE_1] [FILE_2] ···

或者

vim - on [FILE 1] [FILE 2] ...

其中参数说明如下。

(1) -O(大写的 O): 垂直分割(vertical),不同窗口切换用 Ctrl+W+←键或者 Ctrl+W+→键。

(2) -o(小写的 o): 水平分割(horizontal,默认值),不同窗口切换用 Ctrl+W+ ↑ 键或者 Ctrl+W+ ↓ 键。

(3) n: 表示分几个屏,可省略,默认按后面要分割的文件数来决定分几个屏。

(4) [FILE_1] [FILE_2] …: 需要分屏打开的文件。需要注意的是,如果只打开了两 个文件,则重复按两次 Ctrl+W 键即可切换窗口。

图 3-10 所示为使用 vim -O test.txt /etc/passwd /etc/profile 命令所打开的不同文件 垂直分屏的界面。

2. 单个文件分屏编辑

在使用 vim test.txt 编辑文件时,在一般模式下,输入":sp",然后按 Enter 键,会将该文件显示在两个窗口中,实现水平分屏的功能,如图 3-11 所示。

在使用 vim test.txt 编辑文件时,在一般模式下,输入":sp /etc/profile",然后按 Enter 键,会将文件 profile 显示在第一个窗口中,test.txt 显示在第二个窗口中,实现不同文件水 平分屏的功能。如图 3-12 所示。

可以使用 Ctrl+W+ ↑键或者 Ctrl+W+ ↓键在上下两个窗口中切换。

• 86 •

	root@sjh:~	×
文件(F) 编辑(E) 查看(V) 搜索(S) :	终端(T) 帮助(H)	
This is a test file. And this is the first time to use "vim".	root:x:0:0:root:/root:/bin/ba # /etc/profile sh bin:x:1:1:bin:/bin:/sbin/nolo # System wide environment ar	nd
It's easy to use it. I like using "vim",do you lik	gin startup programs, for login daemon:x:2:2:daemon:/sbin:/sb setup	n
e it? There are three modes in vim <mark>.</mark>	in/nologin	in
~	nologin lp:x:4:7:lp:/var/spool/lpd:/s # It's NOT a good idea to ch bin/nologin nge this file unless you kno	ha ow
~~ ~~	<pre>sync:x:5:0:sync:/sbin:/bin/sy what you nc</pre>	er 1
~	<pre>:/sbin/shutdown script in halt:x:7:0:halt:/sbin:/sbin/h # /etc/profile.d/ to make cu</pre>	us
~	alt tom changes to your environm mail:x:8:12:mail:/var/spool/mint, as this ail:/shin/pologin# will prevent the peed for	me
nu Nu	operator:x:11:0:operator:/roo erging in future updates. t:/sbin/nologin	
nu 14	<pre>games:x:12:100:games:/usr/gam pathmunge () { es:/sbin/nologin</pre>	
test.txt 5,29 全部	/sbin/nologin ;; /etc/passwd 1,1 顶端 /etc/profile 1,1 顶结	端

图 3-10 多个文件垂直分屏



图 3-11 单个文件水平分屏

3. 关闭分屏

要取消其他分屏,保留当前分屏,只需要输入":only",然后按 Enter 键即可;或者按 Ctrl+W+o 键也可以关闭其他分屏。

如果要退出当前所在的分屏,输入":q",然后按 Enter 键就可以了。

3.3.3 多文件编辑

可以使用 vim file1 file2 file3 …的方式在 vim 后面同时接好几个文件来开启多文件同





图 3-12 不同文件水平分屏

时编辑的功能。在进行多文件编辑时,可以使用的命令如下。

- (1):n编辑下一个文件。
- (2):N编辑上一个文件。
- (3):files列出目前 vim 开启的所有文件。当执行命令

vim test.txt /etc/passwd /etc/profile

时,屏幕中出现的是第一个文件 test.txt,输入":n",然后按 Enter 键,屏幕上出现 /etc/passwd文件,再输入":n",然后按 Enter 键,屏幕上出现/etc/profile文件。此时,再输入":n",屏幕最后一行将会报错,提示无法切换,已是最后一个文件,如图 3-13 所示。

			root@sjh:~						
文件(F)	编辑(E)	查看(V)	搜索(S)	终端(T)	帮助(H)				
# /etc/	profile								
# Syste # Funct	m wide ions an	environ d alias	ment an es go i	d startu n /etc/l	up programs, for login setup bashrc				
# It's # are c # /etc/ # will	NOT a g loing. I profile prevent	ood ide t's muc .d/ to the ne	a to ch h bette make cu ed for i	ange th: r to cre stom cha merging	is file unless you know what you eate a custom.sh shell script in anges to your environment, as this in future updates.				
pathmur cas	ge () { e ":\${P	ATH}:"	in						
	:"\$1" ;;	:)							
) if	["\$2" PATH=	= "aft \$PATH:\$	er"]; 1	then				
	el	se PATH=	\$1:\$PAT	н					
E165: 🗦	fi E法切换	,已是最	后一个	文件	1,1	顶端			

图 3-13 切换到最后一个文件

此时,如果要切换回第一个文件 test.txt,只能输入":N",然后按 Enter 键,先切换到 /etc/passwd,再输入":N",按 Enter 键,才能切换回 test.txt。

3.3.4 块选择

在使用 vim 编辑一个文件的时候,还可以选择某些字符、某些行或者以长方形的方式 选择资料,然后将选中的文本删除或者复制。此时,可以使用的按键如下。

(1) v: 字符选择,会将光标经过的地方反白选择。按 v 键后,窗口最下方显示出"--可视--"标记。

(2) V: 行选择,会将光标经过的行反白选择。按 V 键后,窗口最下方显示出"--可视行 --"标记。

(3) Ctrl+v: 区块选择,可以用长方形的方式选择资料。按 Ctrl+v 键后,窗口最下方显示出"--可视 块 --"标记。

(4) y: 将反白的地方复制起来。

- (5) d: 将反白的地方删除。
- (6) p: 将复制的内容粘贴。

下面举例说明区块选择的用法,具体步骤如下:

(1) 打开终端,输入命令"vim /etc/netconfig",按 Enter 键。在打开的 vim 中输入":set nu", 按 Enter 键,设置好行号,如图 3-14 所示。

			P	oot@sjh:~				×
文件(F) 编辑(E)	查看(V) 搜索(S)	终端(T) 帮助(H)				
1 1 2 # 3 # 4 # 5 # 6 # 7 # 8 # 9 # 10 #	The netwo conjunct: Entries o contries o The <dev:< td=""><td>ork configurat ion with the T consist of: etwork_id> <sen <device> ice> and <name tation</name </device></sen </td><td>ion fi I-RPC mantic <name toaddr</name </td><td>le. This [.] code in th s> <flags: toaddr_lin _libs> fio</flags: </td><td>file ne li > <pr >s> ≥lds</pr </td><td>is current btirpc lib otofamily> are always</td><td>ly only rary. <proto empty</proto </td><td>y used in name> \ in this</td></dev:<>	ork configurat ion with the T consist of: etwork_id> <sen <device> ice> and <name tation</name </device></sen 	ion fi I-RPC mantic <name toaddr</name 	le. This [.] code in th s> <flags: toaddr_lin _libs> fio</flags: 	file ne li > <pr >s> ≥lds</pr 	is current btirpc lib otofamily> are always	ly only rary. <proto empty</proto 	y used in name> \ in this
12 # 13 u 14 t 15 u 16 t 17 r 18 T 19 u ~	t udp tcp udp6 tcp6 rawip Local unix	tpi_clts tpi_cots_ord tpi_clts tpi_cots_ord tpi_raw tpi_cots_ord tpi_cots_ord	v v v -	inet inet6 inet6 inet loopback loopback	udp tcp udp tcp -	-		
~ :set	nu					1,	1	全部

图 3-14 为文件设置好行号

(2) 按13G 键,将光标移至第13行行首,然后按11→键,将光标移至"tpi_clts"的第一个"t"上,然后按Ctrl+v键,此时窗口最下方显示出"--可视块--"标记,如图3-15所示。

(3) 连续按→键 11次,再按 ↓键 6次,将会选择一个如图 3-16 所示的区块。

(4) 按y键,窗口最下方显示7行的区块内容被复制,如图 3-17 所示。

				root@sjh:~				×
文作	牛(F) 编辑(E) 查看(V) 搜索(S)	终端	(T) 帮助(H)				
1	# # The ne	twork configurat	ion f	ile. This	file i	s curre	ently only	used in
3	# conjun	ction with the T	I-RPC	code in t	he lib	tirpc 1	library.	
4	# # Entrie	s consist of:						
6	#	<pre><network id=""> <se< pre=""></se<></network></pre>	manti	cs> <flags< td=""><td>> <pro< td=""><td>tofamil</td><td>v> <proto< td=""><td>name> \</td></proto<></td></pro<></td></flags<>	> <pro< td=""><td>tofamil</td><td>v> <proto< td=""><td>name> \</td></proto<></td></pro<>	tofamil	v> <proto< td=""><td>name> \</td></proto<>	name> \
8	#	<device></device>	<nan< td=""><td>netoaddr_li</td><td>bs></td><td>cortainz</td><td>cy proce</td><td>1</td></nan<>	netoaddr_li	bs>	cortainz	cy proce	1
9	# # The <d< td=""><td>evice> and <name< td=""><td>toadd</td><td>lr_libs> fi</td><td>elds a</td><td>re alwa</td><td>ays empty</td><td>in this</td></name<></td></d<>	evice> and <name< td=""><td>toadd</td><td>lr_libs> fi</td><td>elds a</td><td>re alwa</td><td>ays empty</td><td>in this</td></name<>	toadd	lr_libs> fi	elds a	re alwa	ays empty	in this
11	<pre># implem #</pre>	entation.						
13	"udp	tpi_clts	v	inet	udp	-	-	
14	tcp	tpi_cots_ord	v	inet	tcp			
15	udp6	tpi_clts	v	inet6	udp	-	· · · ·	
16	tcp6	tpi_cots_ord	v	inet6	tcp	() <u>1</u>	(1	
17	rawip	tpi_raw	-	inet	-	-	-	
18	local	tpi_cots_ord	75	loopback	π.		3. 	
19	unix	tpi_cots_ord	-	loopback	-	-		
- i	可视块 -	-			1x1		13.12	全部

图 3-15 窗口最下方出现"-- 可视 块 --"标记

				root@sjh:~				×
文作	牛(F) 编辑(I	E) 查看(V) 搜索(S)	终端	i(T) 帮助(H)				
1	#							
2	# The ne	twork configurat	ion	file. This	file i	s curr	ently only	used in
3	# conjun	ction with the T	I-RP	c code in t	ne lib	tirpc	library.	
4	#							
5	# Entrie	s consist of:						
6	#							
7	#	<network_id> <se< td=""><td>mant:</td><td>ics> <flags< td=""><td><pro> <pro< pro=""></pro<></pro></td><td>tofami</td><td>ly> <proto< td=""><td>name> \</td></proto<></td></flags<></td></se<></network_id>	mant:	ics> <flags< td=""><td><pro> <pro< pro=""></pro<></pro></td><td>tofami</td><td>ly> <proto< td=""><td>name> \</td></proto<></td></flags<>	<pro> <pro< pro=""></pro<></pro>	tofami	ly> <proto< td=""><td>name> \</td></proto<>	name> \
8	#	<device></device>	<nar< td=""><td>metoaddr_li</td><td>os></td><td></td><td></td><td></td></nar<>	metoaddr_li	os>			
9	#							
10	# The <d< td=""><td>evice> and <name< td=""><td>toad</td><td>dr_libs> fi</td><td>elds a</td><td>re alw</td><td>ays empty</td><td>in this</td></name<></td></d<>	evice> and <name< td=""><td>toad</td><td>dr_libs> fi</td><td>elds a</td><td>re alw</td><td>ays empty</td><td>in this</td></name<>	toad	dr_libs> fi	elds a	re alw	ays empty	in this
11	<pre># implem</pre>	entation.						
12	#			1				
13	udp	tpi_clts	v	inet	udp	-	-	
14	tcp	tpi_cots_ord	V	inet	tcp		-	
15	udp6	tpi_clts	V	inet6	udp	-		
16	tcp6	tpi_cots_ord	v	inet6	tcp	-	-	
17	rawip	tpi_raw	÷.	inet	1	-	-	
18	local	tp1_cots_ord	75	Loopback	20	-		
19	unix	tpi_cots_or <mark>d</mark>	-	loopback	-	-		
	T AD 1t				7.1	2	10 22	人立四
	可恍状-				/ X1	2	19,23	生部

图 3-16 选中区块

(5) 按 G 键,将光标移至文件最后一行,按 o 键,在最后一行的下方新增一行,再按 Esc 键,最后按 p 键,复制的内容将会粘贴在文件的最后一行,如图 3-18 所示。

(6) 由于本例只是演示,所以退出时请输入":q!",按 Enter 键不保存退出。

【试一试】 如果不小心按错了键,只是想让"-- 可视 块 --"标记消失,需要怎么操作? 答案是,按两次 Esc 键。

				root@sjh:~				×
文件	牛(F) 编辑(E) 查看(V) 搜索(S)	终端	(T) 帮助(H)				
1	#							
2	# The ne	twork configurat	ion f	file. This	file i	s curre	ntlv onl	v used in
3	# coniun	ction with the T	I-RPC	code in t	ne lib	tirpc l	ibrarv.	
4	#							
5	# Entrie	s consist of:						
6	#							
7	#	<network_id> <se< td=""><td>manti</td><td>Lcs> <flags< td=""><td><pro>> <pro< pro=""></pro<></pro></td><td>tofamil</td><td>y> <prot< td=""><td>oname> \</td></prot<></td></flags<></td></se<></network_id>	manti	Lcs> <flags< td=""><td><pro>> <pro< pro=""></pro<></pro></td><td>tofamil</td><td>y> <prot< td=""><td>oname> \</td></prot<></td></flags<>	<pro>> <pro< pro=""></pro<></pro>	tofamil	y> <prot< td=""><td>oname> \</td></prot<>	oname> \
8	#	<device></device>	<nan< td=""><td>netoaddr_li</td><td>os></td><td></td><td></td><td></td></nan<>	netoaddr_li	os>			
9	#							
10	# The <d< td=""><td>evice> and <name< td=""><td>toadd</td><td>dr_libs> fie</td><td>elds a</td><td>re alwa</td><td>ys empty</td><td>in this</td></name<></td></d<>	evice> and <name< td=""><td>toadd</td><td>dr_libs> fie</td><td>elds a</td><td>re alwa</td><td>ys empty</td><td>in this</td></name<>	toadd	dr_libs> fie	elds a	re alwa	ys empty	in this
11	<pre># implem</pre>	entation.						
12	#							
13	udp	t pi_clts	v	inet	udp	-	-	
14	tcp	tpi_cots_ord	V	inet	tcp	100		
15	udp6	tpi_clts	v	inet6	udp	-	-	
16	tcp6	tpi_cots_ord	v	inet6	tcp	() <u>-</u>)	(<u>_</u>)	
17	rawip	tpi_raw	-	inet	7	-	-	
18	local	tpi_cots_ord	72	loopback	75	100		
19	unix	tpi_cots_ord	-	loopback	-	-	-	
1.		9					12 12	人言
10	CK OT / L	ines yanked					13,12	全部

图 3-17 复制区块内容

				roc	ot@sjh:~				×
文件	·(F) 编辑(E)	查看(V)	搜索(S)	终端(T)	帮助(H)				
4	#								1
5	# Entries	consist	of:						
6	#								
7	# <r< td=""><td>network_i</td><td>d> <sen< td=""><td>nantics></td><td><flags></flags></td><td><pro< td=""><td>tofamily</td><td>/> <proto< td=""><td>oname> \</td></proto<></td></pro<></td></sen<></td></r<>	network_i	d> <sen< td=""><td>nantics></td><td><flags></flags></td><td><pro< td=""><td>tofamily</td><td>/> <proto< td=""><td>oname> \</td></proto<></td></pro<></td></sen<>	nantics>	<flags></flags>	<pro< td=""><td>tofamily</td><td>/> <proto< td=""><td>oname> \</td></proto<></td></pro<>	tofamily	/> <proto< td=""><td>oname> \</td></proto<>	oname> \
8	#	< d	evice>	<nameto< td=""><td>addr_li</td><td>)5></td><td></td><td></td><td>1997 - 19</td></nameto<>	addr_li)5>			1997 - 19
9	#								
10	# The <dev< td=""><td>/ice> and</td><td><namet< td=""><td>toaddr_1</td><td>ibs> fie</td><td>elds a</td><td>re alway</td><td>/s empty</td><td>in this</td></namet<></td></dev<>	/ice> and	<namet< td=""><td>toaddr_1</td><td>ibs> fie</td><td>elds a</td><td>re alway</td><td>/s empty</td><td>in this</td></namet<>	toaddr_1	ibs> fie	elds a	re alway	/s empty	in this
11	<pre># implemer</pre>	ntation.							
12	#								
13	udp	tpi_clt	S	v j	inet	udp	100		
14	tcp	tpi_cot	s_ord	v i	inet	tcp			
15	udp6	tpi_clt	S	v j	inet6	udp	1	1045	
16	tcp6	tpi_cot	s_ord	v j	inet6	tcp	-	-	
17	rawip	tpi_raw		- j	inet	7.0			
18	local	tpi_cot	s_ord	- 1	oopback	-	380	-	
19	unix	tpi_cot	s_ord	- 1	oopback	÷	() <u>_</u>)	-	
20	tpi_clts								
21	tpi_cots_c	ord							
22	tpi_clts								
23	tpi_cots_c	ord							
24	tpi_raw								
25	tpi_cots_c	ord							*******
多了	6 行						2	10,1	75%

图 3-18 粘贴区块内容

综合实践 3

本章的综合实践以 CentOS 8.1 中的/etc/profile 为例。具体内容包括 15 个步骤,如下 所示。

1. 在/tmp 这个目录下建立一个名为 vimtest 的目录。

2. 进入 vimtest 这个目录当中。

3. 将/etc/profile 复制到本目录下。

4. 使用 vim 打开本目录下的 profile 这个文件。

5. 在 vim 中设定一下行号。

6. 移动到第 37 行,向右移动 15 个字符,观察看到的双引号内是什么数字。

7. 移动到第一行,并且向下搜寻一下"else"这个字符串,观察它在第几行。

8. 将 11~50 行的小写"bin"字符串改为大写"BIN"字符串,并且一个一个挑选是否需要修改,如何下达指令?如果在挑选过程中一直按 y,观察结果会在最后一行出现改变了几个"bin"。

9. 若修改完后要全部复原,有哪些方法?

10. 复制第 11~22 行的内容,并且粘贴到最后一行之后。

11. 如何删除第 2~10 行的注释数据?

12. 将当前这个文件另存为 profile.test。

13. 删除第18行的第11个字符,观察结果出现的第一个单词是什么。

14. 在第一行上面新增一行,该行内容输入"My name is Sheng Jianhui and my number is 2019001001."。

15. 保存后离开 vim。

单元测验3

一、单选题

1. 在 vim 中实现文件	牛多窗口编辑的命令	令是()。			
A. :sp	B. :set nu	C. :set nonu	D. :N		
2. 在 vim 的指令模式	式中,删除光标所在	行的命令是()) 。		
А. уу	B. dd	С. р	D. x		
3. 在 vim 中,可以按	()键进行区块	选择,可以用长方,	形的方式选择资料。		
A. v	B. V	С. р	D. Ctrl+v		
4. 在 vim 的指令模式	式中,使用(),	会将光标所在位置	后的10个字符删除。		
А. 10уу	B. 10dd	C. 10X	D. 10x		
5. 在 vim 的指令模式	式中,使用(),名	会将从光标所在行	开始的 10 行复制。		
А. 10уу	B. 10dd	C. 10G	D. 10X		
二、判断题					
1. 在 vim 中,从编辑	模式可以直接进入	.命令行模式。		()
2. 在 vim 中,从一般	模式可以进入编辑	模式,也可以进入1	命令行模式。	()
3. 在 vim 中,使用":	"可以从一般模式	进入命令行模式。		()
4. 在 vim 中可以对方	て件进行多屏幕编辑	揖。		()
5. 可以使用 vim 后面	面同时接好几个文件	牛来开启多文件同时	时编辑的功能。	()
6. vim 和 vi 都是文本	下编辑器,它们没有	任何区别。		()
7. 使用 vim filename	。命令时,无论什么	时候都不会报错。		()
8. vim 是 vi 的增强版	反,可以用彩色显示	式字,还可以对 S	hell 脚本、C 语言程序	进行	简

• 92 •

单的语法检查。
 9. 离开 vim 后还得要自行删除.filename.swp 才能避免每次打开文件 filename 都会出现警告。
 10. 在 vim 的一般模式下,使用 1G 或者 gg 都可以将光标移至第 1 行。

三、简答题

1. vim 的工作模式有几种?如何进行不同模式之间的切换?

2. vim 的 3 种模式下分别有哪些功能?

3. vim 意外关闭,如何恢复没有保存的文档?

4. vim 的高级功能有哪些?