

# Linux下使用Vim编辑Shell脚本

---

Alex / 2016-10-30 / [free\\_learner@163.com](mailto:free_learner@163.com) / [learning-archive.org](http://learning-archive.org)

更新于2023-05-03，主要是文字排版上的更新，内容基本保持不变。

## 一、背景

---

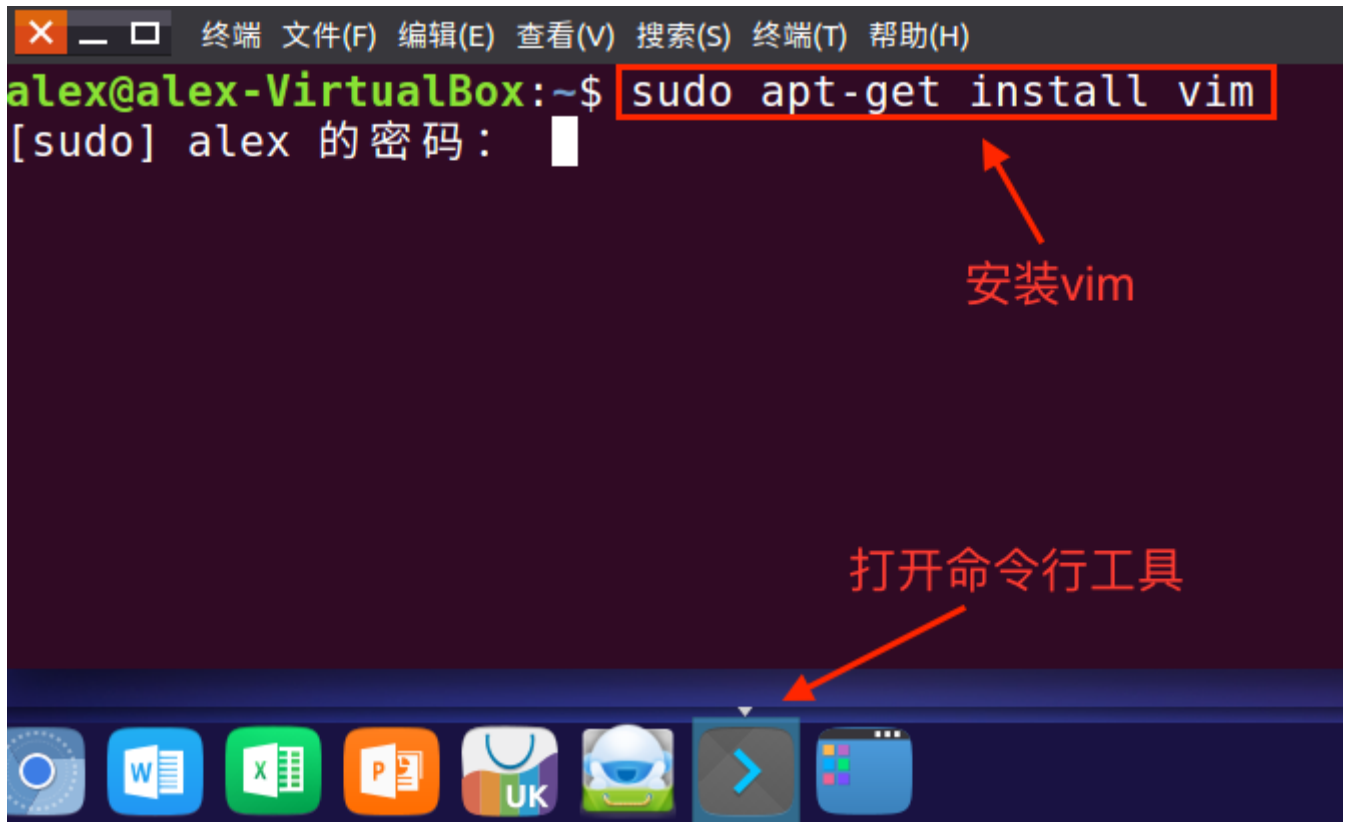
在fMRI数据分析中，由于处理步骤较多，通常的做法是，将许多不同的命令写入一个脚本文件中，运行脚本来完成各种分析。这样的好处是，当我们需要对不同数据进行相同处理的时候，只需要简单修改脚本即可；处理结果异常时，可以通过检查脚本，找出是哪个步骤出了错。所谓脚本，（我简单地理解为）就是将命令以逻辑化的方式组织起来，以完成特定的功能。处理fMRI数据的一般场景是在Linux的Bash Shell中，因此这里的脚本指的是Bash Shell脚本。

Vim就是一款常用的文本编辑器，可以实现在命令行中编辑脚本。特别是当我们需要远程登录服务器处理数据的时候（没有图形界面），Vim是特别方便的。

## 二、安装Vim

---

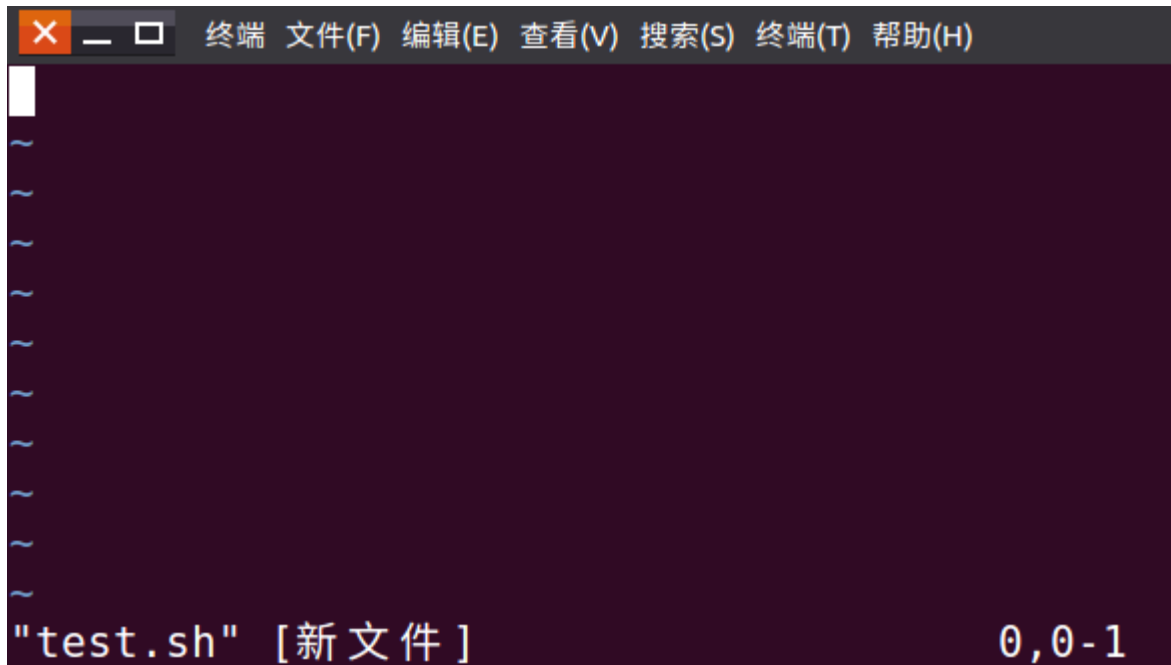
我使用的Linux版本是Ubuntu，需要先安装Vim。打开命令行工具，运行 `sudo apt-get install vim` 由于安装软件需要超级用户权限，因此需要使用sudo命令。Linux系统中允许多个用户存在，不同用户往往有不同的权限。使用sudo命令，会提示输入密码（注意密码不会显示，不要以为是键盘坏了）。



### 三、使用Vim新建或打开脚本

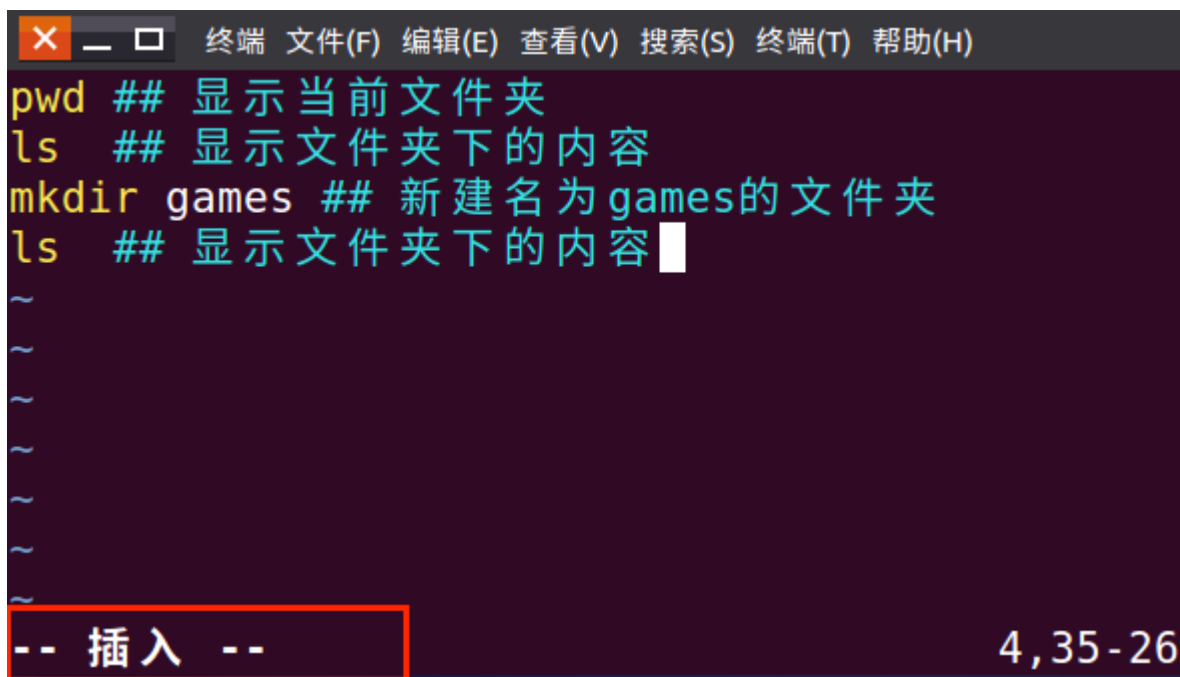
在命令行中执行：`vim test.sh`，如果名为 `test.sh` 的文件已存在就打开它，如果不存在就新建一个名为 `test.sh` 的文件。文件中.sh并没有特殊的意义，仅仅是一种命名习惯而已，表示这是一个Bash Shell脚本。

```
alex@alex-VirtualBox:~$ vim test.sh
```



## 四、使用Vim编辑脚本

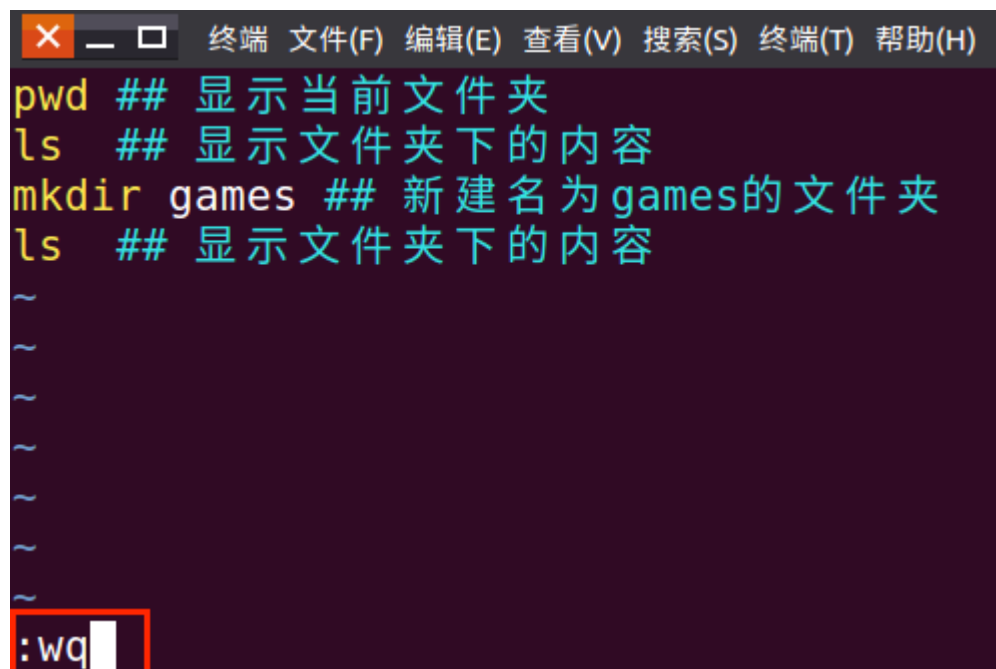
用Vim打开或新建文件后，按下键盘上的 `i` 键即进入了编辑模式。输入下图中简单的内容，其中#后面的内容表示是注释内容，方便自己理解思路，并不会被执行。



## 五、使用Vim保存并退出

输入完后，按下键盘上左上角的 `Esc` 键退出编辑模式，然后输入 `:w` 并回车（w就是写入的意思）。这样就保存了刚才输入的内容。如果要退出Vim编辑器，则输入 `:q` 并回车（q就是退出的意思）。

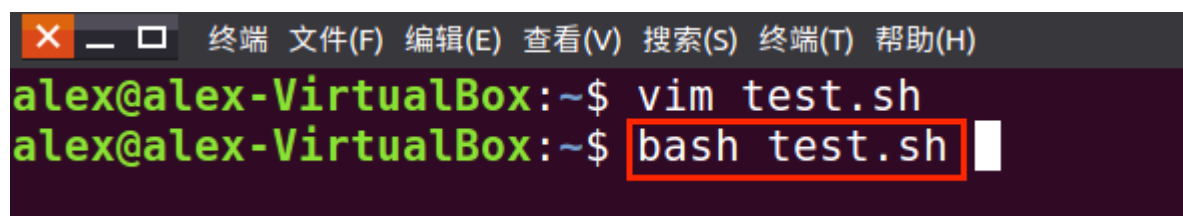
意思)。当然也可以输入 `:wq` 并回车，这样就同时保存了输入的内容并退出Vim编辑器。如果你并不想保存内容，只想退出Vim编辑器，则输入 `:q!` 并回车（!是强制的意思，之所以需要强制，是因为已经输入了内容却不保存）。



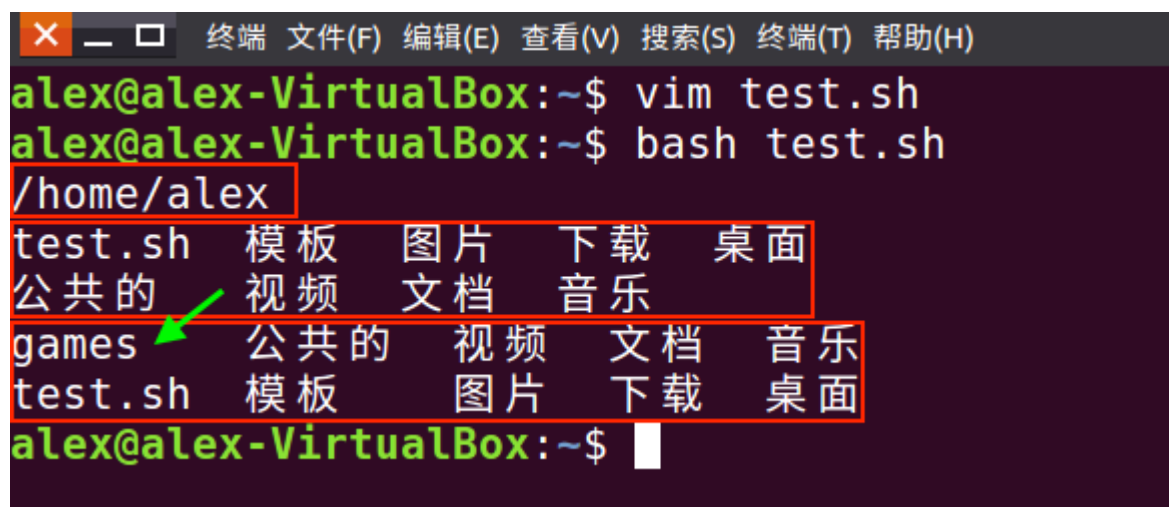
```
终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
pwd ## 显示当前文件夹
ls ## 显示文件夹下的内容
mkdir games ## 新建名为games的文件夹
ls ## 显示文件夹下的内容
~
~
~
~
~
~
~
:wq
```

## 六、运行Shell脚本

编辑好脚本以后，运用以下脚本来看看是否完成了预期的效果。运行脚本的方法之一是在命令行中输入：`bash test.sh` 并回车，其中 `test.sh` 就是刚才我们编辑好的脚本。



```
终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
alex@alex-VirtualBox:~$ vim test.sh
alex@alex-VirtualBox:~$ bash test.sh
```



```
终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
alex@alex-VirtualBox:~$ vim test.sh
alex@alex-VirtualBox:~$ bash test.sh
/home/alex
test.sh 模板 图片 下载 桌面
公共的 视频 文档 音乐
games 公共的 视频 文档 音乐
test.sh 模板 图片 下载 桌面
alex@alex-VirtualBox:~$
```

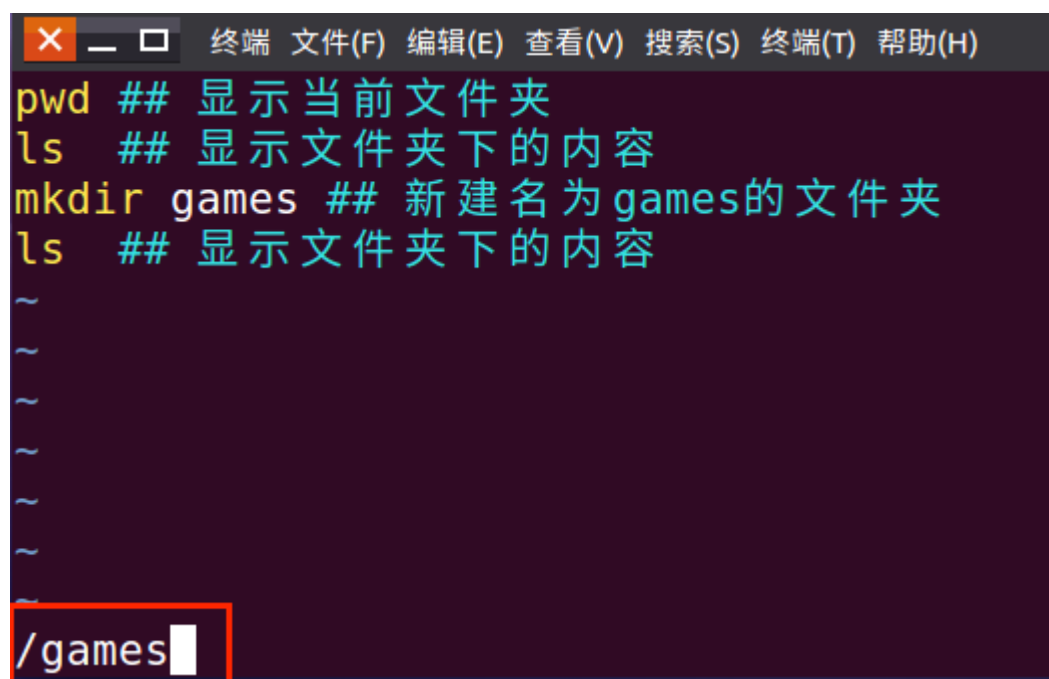
在脚本中我们共运行了4条命令，首先是显示当前文件夹，然后显示文件夹下的内容，然后新建一个名为games的文件夹，最后再次显示文件夹下的内容。从图中可以看到，当前文件夹下确实多了一个名为games的文件夹。

## 七、常用功能

Vim编辑器功能强大，但通常我只需要非常简单的功能即可完成日常数据分析工作。下面介绍我常用的几个功能（注意：不要按i键进入编辑模式，如果进入了编辑模式，按Esc键退出）。

### 1. 查找字符串：

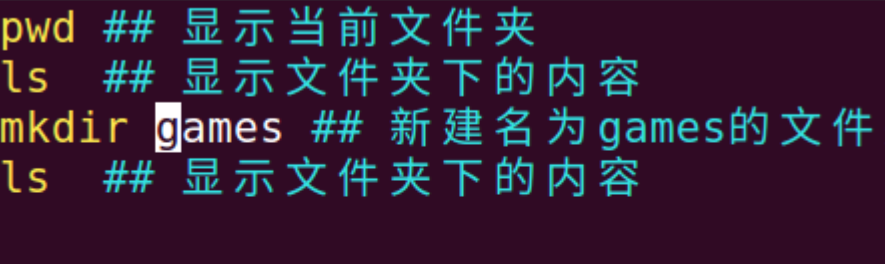
这个功能可用于当我们的脚本很长，而我们需要查找一个字符串的时候。打开一个脚本，比如 `vim test.sh`，输入 `/games` 并回车，这时候光标就跳到了第一次出现games的位置。如果脚本中不止一个games，按 `n` 键光标跳到下一个结果的位置。

A terminal window with a dark purple background and light blue text. The window title bar shows icons for a window, a terminal, and a file, followed by the text '终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)'. The terminal content shows the following commands and their outputs:   
1. `pwd` followed by `## 显示当前文件夹`   
2. `ls` followed by `## 显示文件夹下的内容`   
3. `mkdir games` followed by `## 新建名为games的文件夹`   
4. `ls` followed by `## 显示文件夹下的内容`   
Below these, there are several lines of tilde (~) characters representing the output of the `ls` command. At the bottom, the text `/games` is entered, and a white cursor is positioned at the end of the text. A red rectangular box highlights the `/games` text and the cursor.

A terminal window with a dark background and light green text. The title bar at the top shows standard window controls and menu items: "终端", "文件(F)", "编辑(E)", "查看(V)", "搜索(S)", "终端(T)", and "帮助(H)". The command history displayed is as follows:  
`pwd` ## 显示当前文件夹  
`ls` ## 显示文件夹下的内容  
`mkdir games` ## 新建名为games的文件夹  
`ls` ## 显示文件夹下的内容  
~  
~  
~  
~  
~  
~  
~  
The prompt character is a tilde (~). At the bottom left, the path `/games` is partially visible. In the bottom right corner, the page number `3,7` is shown.

## 2. 显示行数:

脚本运行过程中如果报错，一般会返回哪一行出现了错误，这时我们需要返回到脚本中的特定行去纠错。这时脚本中能显示行数可以方便定位。用Vim打开脚本，输入 `:set nu` 并回车，这样脚本就能显示行数了。



```
1 pwd ## 显示当前文件夹
2 ls ## 显示文件夹下的内容
3 mkdir games ## 新建名为games的文件夹
4 ls ## 显示文件夹下的内容

~
~
~
~
~
~
~
~

:set nu 3,7
```

### 3. 回到第一行或最后一行

用Vim打开脚本，按 `gg`（两个g）回到第一行，按 `shift g`（shift键加上g键）回到最后一行。

#### 4. 回到行首或行尾

用Vim打开脚本，按 `0`（数字0）回到行首，按 `$` 回到行尾（常用的键盘需要按shift键加上4才能输出\$符号）。