

39 Vim岂是池中物，宝剑锋从磨砺出

更新时间：2019-08-08 09:41:19



“

人生太短，要干的事太多，我要争分夺秒。

——爱迪生

”

内容简介

1. 前言
2. 安装 Vim
3. Vim 的各种编辑模式
4. 基本操作（移动，写入，保存，等等）
5. 总结

1. 前言

上一课是 [带你玩转Linux和Shell编程 | 第四部分测试题](#)。

终于来到系列课程的第五部分了。

在这个部分里，我们要把前面四部分所学的知识总结起来，并学习新的知识：Shell 脚本。

Shell 脚本是一个整合在 Linux 系统里的“小”编程语言。它是系统的重要元素，我们不得不学习。

然而，要编写程序，我们需要使用一个更配得上“编程”这个名词的文本编辑器。

虽然我们已经学习了 Nano 这个文本编辑器，但是我们也说过，Nano 比较基础，要“傍大腿”就要傍“肥美多汁”的。

因此，这一课我们先来学习编辑器中的神器：Vim。

在 Linux 文本编辑器的江湖中，有两件上古流传的“神器”：

- Vim：Vi 文本编辑器的进阶版，是 Vi iMproved 的缩写，表示“改进了的 Vi”。Vi 非常流行，在大多数基于 Unix 演进的操作系统中一般是默认安装的，Linux 便是其中之一。Vim 从名字即可知道比 Vi 编辑器更加先进。Vim 的官网是 <https://www.vim.org>。
- Emacs：著名的开源先驱 Richard Stallman（GNU 项目的奠基人，我们以前的课已经介绍过）所开发的文本编辑器（后来有很多人参与改进）。一般没有默认安装在 Linux 系统中，但是可以用 apt 等包管理软件轻松安装。Emacs 可以安装很多插件，做很多事情，比如浏览网页、听音乐、看视频等等。基本你想得到的都可以做，除了不能当你女朋友... Emacs 的官网是 <https://www.gnu.org/software/emacs>。

这两件“神器”也引发了追随者的派别之争，经常可以看到这两个门派的粉丝（“噢，我们这是古装剧啊，怎么出现粉丝了。好吧，穿越回现代”）“掐架”，硬要说明自己拥护的编辑器比另一个好，搞得选择文本编辑器像选择信仰一般（是的，我知道，这些 crazy programmers（疯狂的程序员）。我本人可是很理智的程序员好嘛～ “真的吗？”...）。

但不可讳言，选择一个自己惯用的文本编辑器确实很重要，因为你需要不断用这个神器“打怪练级”才能提高。一开始要是选错就不好了，万一不小心练了“辟邪剑谱”...

一般我们很少看到有人两种神器都练的。不过我练了两种（“闲得发慌型”），说实话个人更偏爱 Emacs（不过目前工作中还是主要用 Vim，为了配合我同事们，他们基本都用 Vim）。

这一课和下一课我们先介绍 Vim，第三课我们才开始 Shell 之旅。

2. 安装 Vim

Vim 还是 Emacs？Emacs 还是 Vim？好纠结...

但是你会说：“你说了上面这些，对我选择 Vim 还是 Emacs 的决定并没有什么用啊。”

好吧，那我就给一个比较实际的建议：

尽量用你身边的 Linux 高手或同事或同学朋友用得最多的那个文本编辑器。

因为你们经常要交流，而且身边的朋友可以给你更好的使用建议，或者推荐一些技巧和插件什么的。

我任职过的几个公司，大部分同事都是用 Vim 的，我也一直用 Vim。但之前在某个法国公司工作时，技术老总“强制”每个人用 Emacs（我晕... 法国不是崇尚“自由、平等、博爱”吗？），然后我只好去学了 Emacs，并且竟然喜欢上了 Emacs... “哎呀，真香”。所以现在 Vim 和 Emacs 都可以用。

安装并启动 Vim

在大部分 Linux 发行版中，Vim 是默认安装的。但是不保证有个别 Linux 发行版没有默认安装 Vim（比如 Ubuntu 中没有默认安装 Vim）。安装 Vim，只需要在终端中输入以下命令：

```
sudo apt install vim
```

或者：

```
sudo apt-get install vim
```

顺便提一下 apt 和 apt-get 的区别：

在 [这个页面](#) 有以下陈述：

"APT is a vast project, whose original plans included a graphical interface. It is based on a library which contains the core application, and apt-get is the first front end — command-line based — which was developed within the project. apt is a second command-line based front end provided by APT which overcomes some design mistakes of apt-get."

翻译成中文大致意思是：

“APT 是一个大型项目，最初的计划包含了开发一个图形界面。它是基于一个库，这个库包含了核心应用，apt-get 是第一个基于命令行的前端（apt-get 是在 APT 这个项目中被开发的）。apt 是第二个基于命令行的前端，也是在 APT 项目中被开发的，它修正了 apt-get 的一些设计错误。”

上面这段英文，简而言之就是：apt 和 apt-get 大致来说很相似，许多命令也是相同的，不过 apt 比 apt-get 更优一些。

安装完 Vim 之后，我们只需要运行以下命令就可以启动 Vim 了：

```
vim
```

当然了，如果你输入 vi，那么运行的会是 vi 编辑器。但是我们肯定更推荐 vim 啦。

vimtutor : Vim 的教程程序

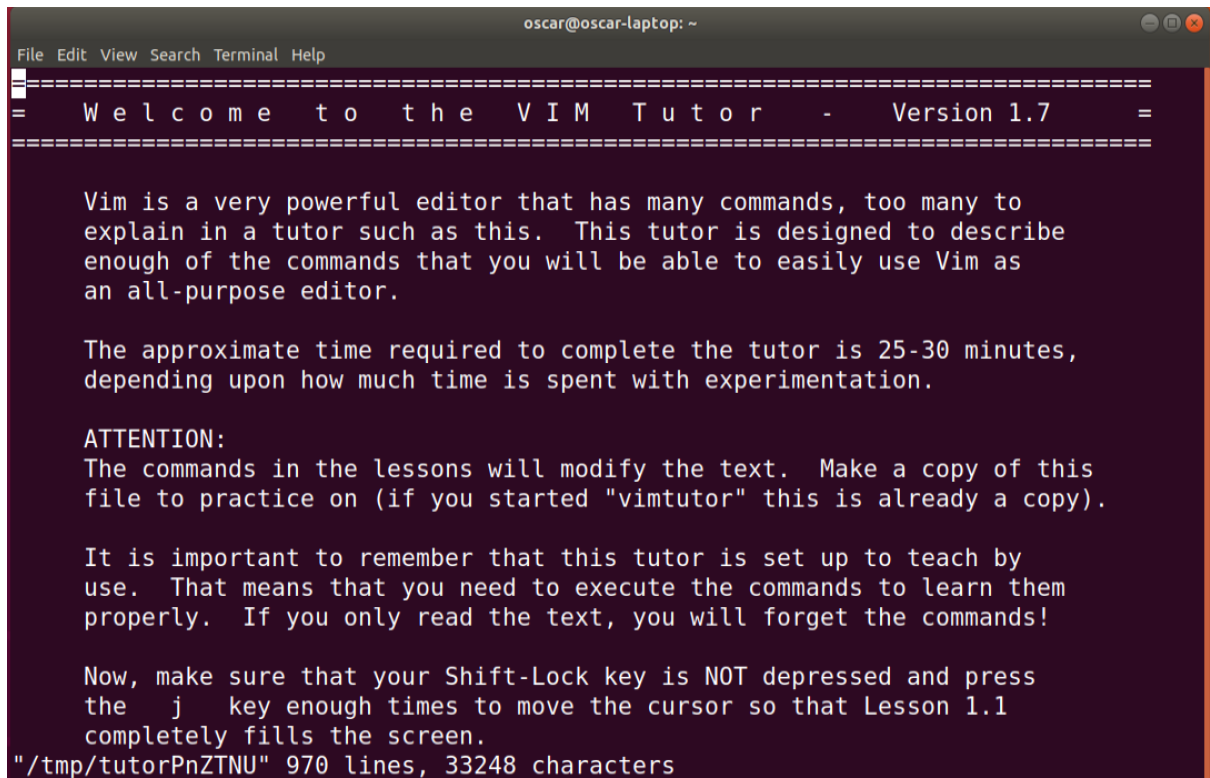
对于新手，Vim 的开发者们也很贴心，它们在 Vim 中内嵌了一个教程小程序。

这个程序只需要输入以下命令就可以执行：

```
vimtutor
```

vimtutor 是 vim 和 tutorial 的缩写，tutorial 是英语“教程”的意思，因此，vimtutor 就是“Vim 教程”的意思。

如果你输入 vimtutor 显示没有此命令，那么你可以安装 vim-common 这个包。不过一般 vimtutor 都已经随 Linux 发行版安装了。运行后如下：



```
oscar@oscar-laptop: ~
File Edit View Search Terminal Help
=====
= Welcome to the VIM Tutor - Version 1.7 =
=====

Vim is a very powerful editor that has many commands, too many to
explain in a tutor such as this. This tutor is designed to describe
enough of the commands that you will be able to easily use Vim as
an all-purpose editor.

The approximate time required to complete the tutor is 25-30 minutes,
depending upon how much time is spent with experimentation.

ATTENTION:
The commands in the lessons will modify the text. Make a copy of this
file to practice on (if you started "vimtutor" this is already a copy).

It is important to remember that this tutor is set up to teach by
use. That means that you need to execute the commands to learn them
properly. If you only read the text, you will forget the commands!

Now, make sure that your Shift-Lock key is NOT depressed and press
the j key enough times to move the cursor so that Lesson 1.1
completely fills the screen.
"/tmp/tutorPnZTNU" 970 lines, 33248 characters
```

如果你英语好，其实基本的 Vim 操作通过这个 vimtutor 的程序已经可以练习并掌握，因为它其实就是运行 vim 程序，然后打开了一个预先定义好的帮助文档。你可以在里面“打怪升级”。

一般跟着这个 vimtutor 练手，大概 30 分钟就可以完成了。完成后你对 Vim 的操作就有了基本认识。当然了，这个教程只是入门，不可能介绍 Vim 的所有元素和可能性。

要退出这个 Vim 教程，只需要输入：

```
:q
```

也就是“冒号 + 小写的 q”（q 是 quit 的首字母，是英语“退出”的意思），再按回车，就退出了。

3. Vim 的各种编辑模式

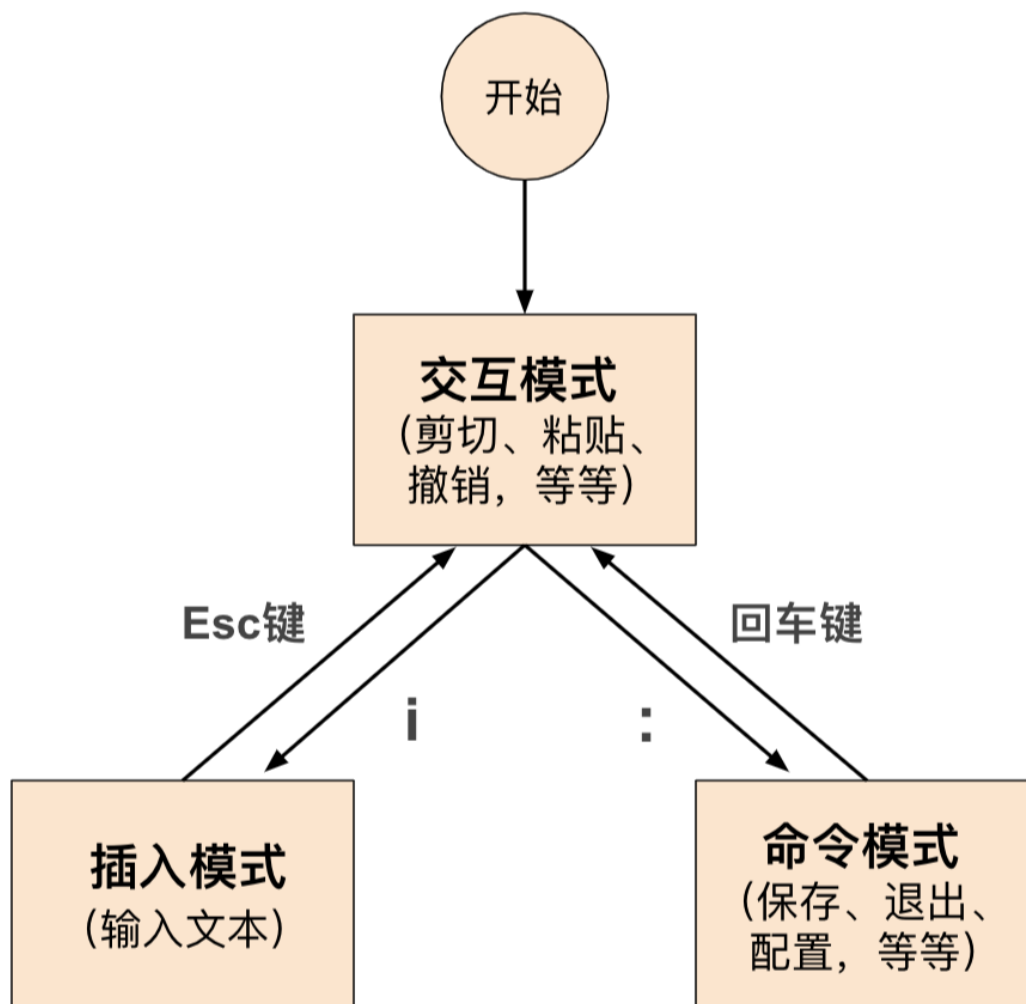
首先我们运行 Vim。之前我们已经演示过了，只需要在命令行输入以下命令：

```
vim
```

Vim 就会乖乖地运行起来，如下图所示：

小结一下：

Vim 拥有三种模式：交互模式、插入模式、命令模式。一开始启动 Vim 的时候，默认是进入交互模式的。这三种模式中，我们最熟悉的肯定是插入模式，这也是一般文本编辑器的模式，比如 Windows 系统的记事本。其它两种模式（交互模式和命令模式），也许我们一下子有点难以接受。可以用下图来概括：



但是，聪明如你也许要发问：“为什么要在一个文本编辑器里插入如此多的模式呢？为什么没有菜单呢？为什么不用一个图形界面的文本编辑器呢？用一个鼠标不是更方便吗？”

以上都是好问题。但我只能先尝试着回答你，当然你也要对我够信任：如果 Vim 的开发创建了这么多看似复杂的模式，那肯定是有其用意的。

实际上，正是有了这三个模式，Vim 才会如此强大。在下面的课中，我们会慢慢展示。

你会发现 Vim 可以实现一般的文本编辑器所不能做的操作，例如：删除当前字符；从光标开始处剪切文本直到行末，将剪贴板中的文本粘贴四次；跳转到第 100 行；跳转到最后一行等等。

以上所列举的所有操作，都只需要键盘就可以，完全不需要鼠标。因此，你也许有一天会感叹：“一入 Vim 深似海，从此鼠标是路人”。但是，一开始并不轻松，因为大部分按键都是需要记住的。

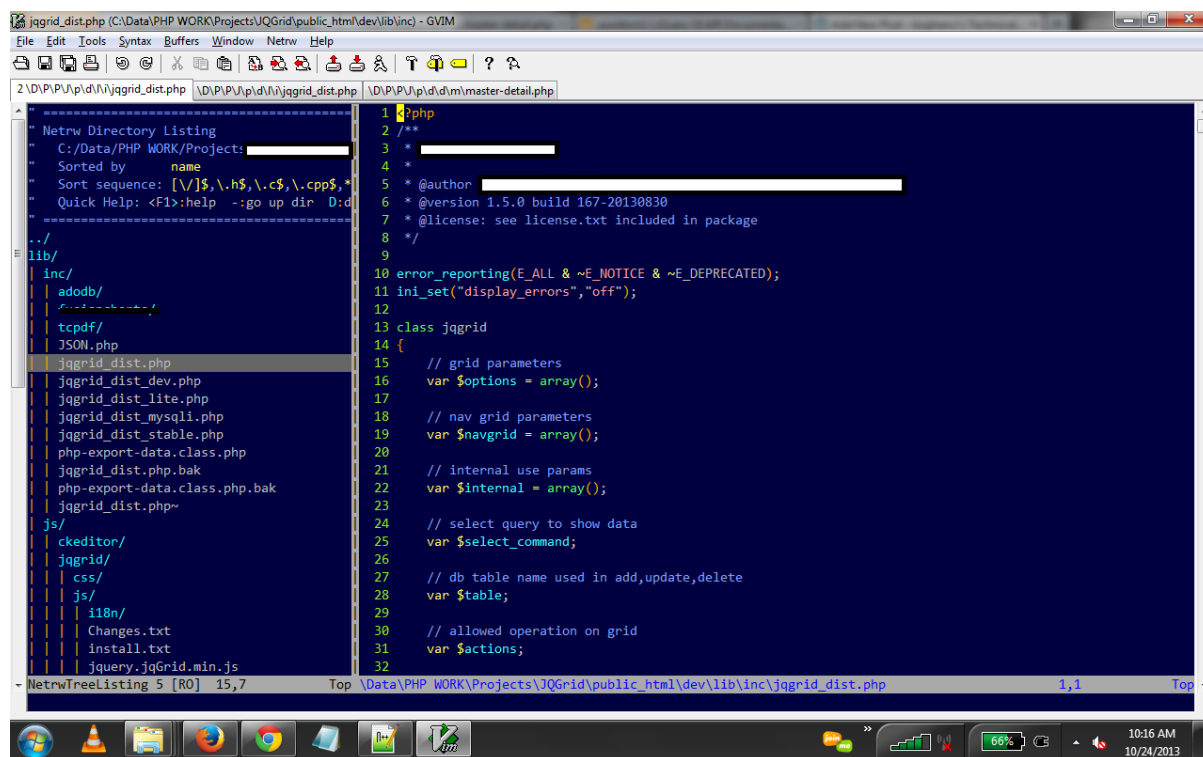
这和学习“盲打”有点像。很多年前，我在电脑上打字是用一根指头并且眼睛得偷瞄键盘。后来学了编程，感觉用一根指头打字实在是太没效率了。

因此，我从网上搜了“盲打”的示意图，逼着自己把十根手指放在键盘的特定位置。一开始练习“盲打”的时候，输入速度比我用“一指指”时还慢。但是渐渐地，就“健指如飞”了，打字效率绝对提高一个数量级。到最后，就达到了“人键合一”的境界，简直就是…



入门 Vim 也是如此。一开始不轻松，但是渐渐地，它会成为你编程的生产力工具，大大提高效率。

当然了，有些人还是偏爱图形界面。Vim 也有一个图形界面的版本，称为 gVim（Gnome Vim 的缩写）。你可以安装 gVim（即使你是在 KDE 桌面系统下）。gVim 的功能和 Vim 是基本一样的。gVim 也有 Windows 版本，如下图所示：



可以看到 gVim 有菜单和工具栏，和一般的文本编辑器类似，但我更偏爱命令行形式的 Vim。

4. 基本操作（移动，写入，保存，等等）

我们会通过几个步骤来一点点认识 Vim。这些步骤会从简单到复杂，但是我还是很厚道的（“真的吗？”…），会慢慢来，不会一下子把你塞成一个“胖子”。

打开 Vim

如果你在终端命令行中输入 `vim`，不添加任何参数，回车后 Vim 就会被运行起来，我们之前也见过其庐山真面目了。

你也可以用 Vim 来打开一个文件，只需要在 `vim` 后面再加文件名，如下：

```
vim 文件名
```

如果文件不存在，那么会被创建。

i：插入文本

我们从一个空白文件开始（就是在命令行输入 `vim`，然后回车）。

首先按下键盘上的字母键 i (小写的 i)，这样我们就从初始的交互模式进入了插入模式，就可以开始键入文本了：

[illegible]

注意看：左下角有一个 `-- INSERT --`，说明我们就处在 Insert Mode（插入模式）中。

你可以在里面输入几行文本，就像我一样，然后按 `Esc` 键，回到交互模式。你会看到左下角的 `-- INSERT --` 消失了，我们就重新回到了交互模式：

在交互模式下：

- 为了将光标定位到一行的开始位置，我们只需要按下数字键 0 即可。当然了，键盘上的“Home”键也有相同效果。但是我们还是建议用 0 键，毕竟不用隔那么老远去按 Home 键。
- 为了将光标定位到一行的末尾，我们只需要按下美元符号键 \$ 即可。当然了，键盘上的“End”键也有相同效果。但是我们还是建议用 \$ 键，毕竟不用隔那么老远去按 End 键。

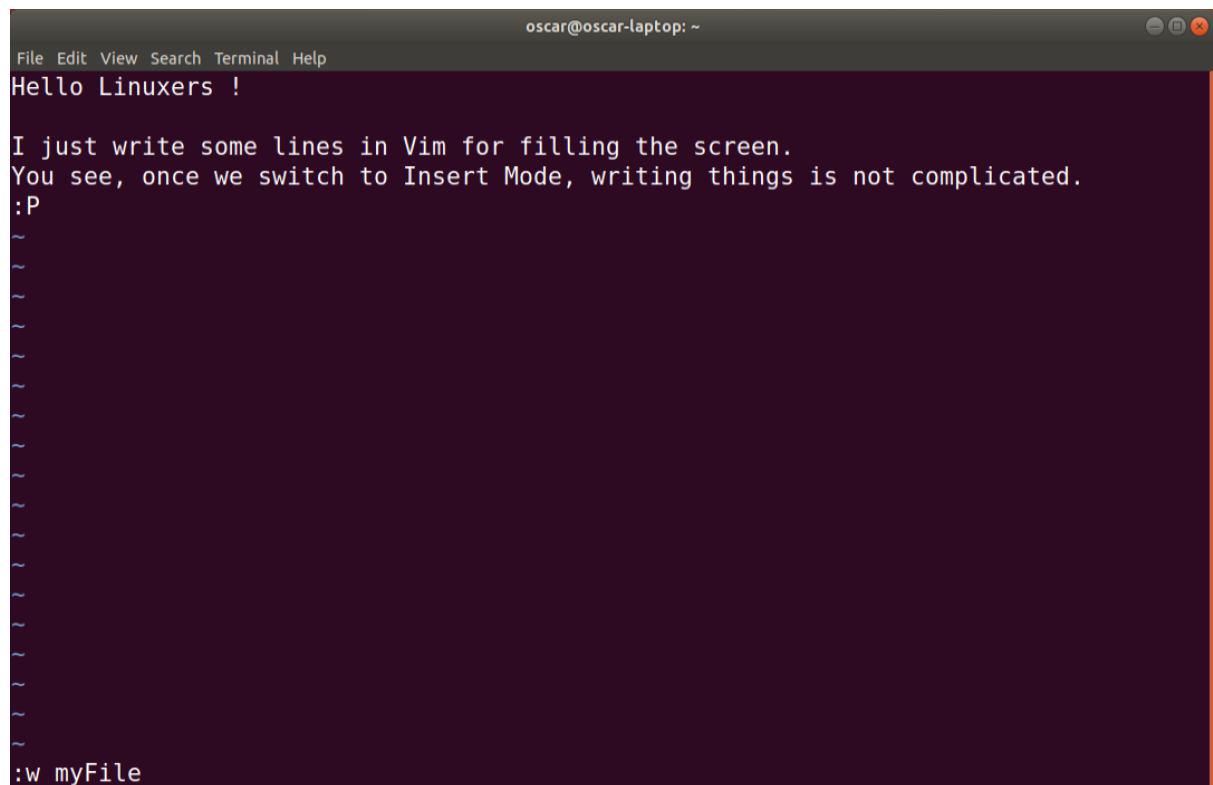
w：一个单词一个单词地移动

在交互模式下，按字母键 w（w 是 word 的首字母，是英语“单词”的意思），我们可以一个单词一个单词地移动，有时候比一个字符一个字符地移动可高效多了。

:w：保存文件

为了保存文件，我们首先要进入交互模式。

按下冒号键（:），以进入命令模式，然后按下 w（w 是 write 的首字母，是英语“写入”的意思）键，再输入你要保存的文件名字，因为目前我们还没有创建文件，只是从 Vim 的交互模式进入插入模式后直接写入的上面那些文本，如下图所示：



```
oscar@oscar-laptop: ~
File Edit View Search Terminal Help
Hello Linuxers !

I just write some lines in Vim for filling the screen.
You see, once we switch to Insert Mode, writing things is not complicated.
:P
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:w myFile
```

可以看到，我给文件起名叫 myFile，是 my 和 file 的缩合，my 是英语“我的”的意思，file 是英语“文件”的意思，因此 myFile 表示“我的文件”。

然后再回车，如下图所示：

欢迎在这里发表留言，作者筛选后可公开显示



目前暂无任何讨论

千学不如一看，千看不如一练