高级功能入门

## 一、多文件编辑

### 1.使用vim编辑多个文件

编辑多个文件有两种形式，一种是在进入vim前使用的参数就是多个文件。另一种就是进入vim后再编辑其他的文件。 同时创建两个新文件并编辑

$ vim 1.txt 2.txt

默认进入1.txt文件的编辑界面

* 命令行模式下输入:n编辑2.txt文件，可以加!即:n!强制切换，之前一个文件的输入没有保存，仅仅切换到另一个文件
* 命令行模式下输入:N编辑1.txt文件，可以加!即:N!强制切换，之前文件内的输入没有保存，仅仅是切换到另一个文件

### 2.进入vim后打开新文件

* 命令行模式下输入:e 3.txt 打开新文件3.txt
* 命令行模式下输入:e# 回到前一个文件
* 命令行模式下输入:ls可以列出以前编辑过的文档
* 命令行模式下输入:b 2.txt（或者编号）可以直接进入文件2.txt编辑
* 命令行模式下输入:bd 2.txt（或者编号）可以删除以前编辑过的列表中的文件项目
* 命令行模式下输入:e! 4.txt，新打开文件4.txt，放弃正在编辑的文件
* 命令行模式下输入:f 显示正在编辑的文件名
* 命令行模式下输入:f new.txt，改变正在编辑的文件名字为new.txt

### 3.恢复文件

如果因为断电等原因造成文档没有保存，可以采用恢复方式，vim -r进入文档后，输入:ewcover 1.txt来恢复

$ vim -r 1.txt

## 二、可视模式

### 1.可视模式命令简介

* 在普通模式下输入v（小写），进入字符选择模式，就可以移动光标，光标走过的地方就会选取。再次按下v会后就会取消选取。
* 在普通模式下输入Shift+v（小写），进入行选择模式，按下V之后就会把整行选取，您可以上下移动光标选更多的行，同样，再按一次Shift+v就可以取消选取。
* 在普通模式下输入 Ctrl+v（小写），这是区域选择模式，可以进行矩形区域选择，再按一次Ctrl+v取消选取。
* 在普通模式下输入d删除选取区域内容
* 在普通模式下输入y复制选取区域内容

### 2.可视模式命令练习

拷贝练习文件到当前目录

$ cp /etc/protocols .

打开练习文件

$ vim protocols

* 在普通模式下9G跳转到第9行,输入Shift+v（小写V），进入可视模式进行行选择，选中5行，按下>>缩进，将5行整体缩进一个shiftwidth
* 在普通模式下输入 Ctrl+v（小写v），进入可视模式进行矩形区域选择，选中第一列字符然后x删除整列

见vim演示7

## 三、视窗操作

### 1.视窗操作简介

vim可以在一个界面里打开多个窗口进行编辑，这些编辑窗口称为vim的视窗。 打开方法有很多种，例如可以使用在命令行模式下输入:new 打开一个新的vim视窗，并进入视窗编辑一个新文件（普通模式下输入Ctrl+w也可以,但是Ctrl+w在chrome下会与chrome关闭标签页的快捷键冲突，所以使用该快捷键你可以在IE或其它浏览器进行练习），除了:new命令，下述列举的多种方法也可以在命令模式或普通模式下打开新的视窗：

* 命令行模式下输入:sp 1.txt 打开新的横向视窗来编辑1.txt
* 命令行模式下输入:vsp 2.txt 打开新的纵向视窗来编辑1.txt
* 普通模式下Ctrl-w s 将当前窗口分割成两个水平的窗口
* 普通模式下Ctrl-w v 将当前窗口分割成两个垂直的窗口
* 普通模式下Ctrl-w q 即 :q 结束分割出来的视窗。如果在新视窗中有输入需要使用强制符！即:q!
* 普通模式下Ctrl-w o 打开一个视窗并且隐藏之前的所有视窗
* 普通模式下Ctrl-w j 移至下面视窗
* 普通模式下Ctrl-w k 移至上面视窗
* 普通模式下Ctrl-w h 移至左边视窗
* 普通模式下Ctrl-w l 移至右边视窗
* 普通模式下Ctrl-w J 将当前视窗移至下面
* 普通模式下Ctrl-w K 将当前视窗移至上面
* 普通模式下Ctrl-w H 将当前视窗移至左边
* 普通模式下Ctrl-w L 将当前视窗移至右边
* 普通模式下Ctrl-w - 减小视窗的高度
* 普通模式下Ctrl-w + 增加视窗的高度

### 2.视窗操作练习

打开练习文件

$ vim 1.txt

* 命令行模式下输入:new 打开一个新的vim视窗
* 命令行模式下输入:sp 2.txt 打开新的横向视窗来编辑2.txt
* 命令行模式下输入:vsp 3.txt 打开新的横向视窗来编辑3.txt
* 如果使用非chrome浏览器可以使用Ctrl+w进行视窗间的跳转
* 分别在不同视窗的命令行模式下输入:q!退出多视窗编辑

见vim演示8

## 四、文档加密

### 1.创建加密文档

$ vim -x file1

输入您的密码 确认密码 这样在下一次打开时，vim就会要求你输入密码

## 五、在vim执行外部命令

在命令行模式中输入!可以执行外部的shell命令

* :!ls 用于显示当前目录的内容
* :!rm FILENAME用于删除名为 FILENAME 的文件
* :w FILENAME可将当前 VIM 中正在编辑的文件另存为 FILENAME 文件

## 六、帮助系统

### 1.vim中的查看帮助

* 普通模式下按F1打开vim自己预设的帮助文档
* 命令行模式下输入:h shiftwidth 打开名为shiftwidth的帮助文件
* 命令行模式下输入:ver 显示版本及参数

## 七、功能设定

### 1.vim的功能设定

可以在编辑文件的时候进行功能设定，如命令行模式下输入:set nu（显示行数），设定值退出vim后不会保存。要永久保存配置需要修改vim配置文件。 vim的配置文件~/.vimrc，可以打开文件进行修改，不过务必小心不要影响vim正常使用

### 2.获取目前的设定

* 命令行模式下输入:set或者:se显示所有修改过的配置
* 命令行模式下输入:set all 显示所有的设定值
* 命令行模式下输入:set option? 显示option的设定值
* 命令行模式下输入:set nooption 取消当期设定值

### 3.set功能的说明

* 命令行模式下输入:set autoindent(ai) 设置自动缩进
* 命令行模式下输入:set autowrite(aw) 设置自动存档，默认未打开
* 命令行模式下输入:set background=dark或light，设置背景风格
* 命令行模式下输入:set backup(bk) 设置自动备份，默认未打开
* 命令行模式下输入: set cindent(cin) 设置C语言风格缩进

更多详细参数请参考vim手册

## 八、总结

通过这四章的简单学习，相应你应该掌握了vim的基本操作和使用，但本课程的主要目的是为了让你在学习实验楼上面其他需要用到vim的课程中不会有任何问题。如果你想单纯的学习并熟练掌握vim编辑器，通过各类教程包括本课程的学习是不能够满足的，因为要熟练掌握是跟你个人的选择有关，这需要你不断的联系并坚持长期使用vim完成各种编辑操作才能达到，同时你还需要掌握如何更改和编写vim的配置文件及安装各类vim插件来实现各种强大的功能满足你的各种苛刻的需求，最后希望你在实验楼玩得愉快