

第1章 VIM编辑器

1.VIM介绍

1. 上古编辑器之神，屠龙利器
2. vim 是从 vi 发展出来的一个文本编辑器。
3. 互联网上有个段子
4. 问：如何打印出真随机数？
5. 答：打开VIM然后让新手退出。

2.VIM特点

1. 学习曲线陡峭,全是键盘操作，需要记忆的快捷键比较多。
2. 如果突破了学习的障碍，非常熟练使用VIM快捷键之后可以达到魔法般的效果。
3. 职场装逼利器，没有之一。

3.VIM工作模式

普通模式

1. 默认处于这种模式，这种模式下只能移动光标上下翻页查看数据，并不能修改数据，可以理解为只读模式。
2. 注意：大部分快捷键都是运行在普通模式下的。

编辑模式

1. 只有进入编辑模式之后才可以修改文本内容。
2. 进入编辑模式需要在普通模式下输入特定的快捷键，不同的快捷键进入的编辑模式效果也不同。
3. 同样，退出编辑模式也需要输入特定的快捷键。

命令行模式

1. 用于接收特定指令的模式，类似于shell命令
2. 主要功能如下：
3. 1. 查找关键词
4. 2. 替换或删除关键词
5. 3. 保存退出等操作

4.VIM生存指南

普通模式：移动

```
1 | 上下左右
2 | h j k l
3 | g g
4 | G
5 | O
6 | $
```

普通模式：复制粘贴

```
1 | yy
2 | nyy
3 | p
```

普通模式：撤销和快进

```
1 | u
2 | ctrl + r
```

普通模式：替换

```
1 | r
```

普通模式：删除

```
1 | dd
2 | n dd
3 | d G
4 | d O
5 | d $
```

普通模式：视图界面

```
1 | ctrl + v + 选择区块
2 | y
3 | p
```

编辑模式：进入编辑模式

```
1 | i
2 | a
3 | I
4 | A
5 | o
6 | O
7 | ESC
```

命令行模式：保存/退出/设置

```
1 | :  
2 | :wq  
3 | :wq!  
4 | :q!  
5 | :set nu
```

命令行模式：查找关键词

普通模式下

```
1 | /关键词  
2 | n      向下查找关键词  
3 | N      向上查找关键词  
4 | :noh   取消关键词高亮显示
```

5.VIM降龙十八掌

快速加减

```
1 | N + Ctrl + A  
2 | N + Ctrl + X
```

快速定位

```
1 | f 单词
```

单词补全

```
1 | 关键词 + Ctrl + p
```

6.VIM修炼心法

```
1 | 1.不要想着一下全部学会，先熟练几个最基本的，用熟悉了再练其他的。  
2 | 2.尽可能的使用快捷键，所有操作之前先想想有没有快捷键。  
3 | 3.少用delete和上下左右键。
```

第2章 文本过滤/查看命令

1.echo：输出文本

重要选项

```
1 | -n  
2 | -E  
3 | -e
```

练习题:

打印文本输出到屏幕里

```
1 | echo "oldboy"
```

将单行文本追加到某个文件中

```
1 | echo "oldboy" >> file1.txt
```

激活特殊符号

```
1 | echo -e "第一行\n第二行"
```

打印系统环境变量

```
1 | echo $PATH
```

打印序列号

```
1 | echo {1..10}
```

单引号和双引号区别

```
1 | 单引号：所见即所得，输入什么就输出什么，特殊符号没有意义
2 | 双引号：可以解析变量
3 |
4 | [root@Centos7-100 ~]# echo '$PATH'
5 | $PATH
6 | [root@Centos7-100 ~]# echo "$PATH"
7 | /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
```

2.cat：合并文件或查看文件内容

重要选项

```
1 | -n
2 | -A
3 | EOF
```

练习题:

查看文件内容

```
1 | cat file1.txt
```

查看文件内容并显示行号

```
1 | cat -n file1.txt
```

查看文件内容的特殊字符

```
1 | cat -A file1.txt
```

将file1.txt的内容复制给file2.txt

```
1 | cat file1.txt > file2.txt
```

将file1.txt和file2.txt的内容合并成file3.txt

```
1 | cat file1.txt file2.txt >> file3.txt
```

将多行文本写入到file4.txt

```
1 | cat > file4.txt << EOF
2 | 123
3 | 456
4 | 789
5 | EOF
```

将多行文本写入到file4.txt，但是不解析变量内容

```
1 | cat > file5.txt << 'EOF'
2 | $PATH
3 | 123
4 | $PATH
5 | 456
6 | EOF
```

3.head：显示文件头部信息

重要选项

```
1 | -n
```

练习题：

显示文件前10行

```
1 | head /etc/passwd
```

显示文件前3行

```
1 | head -3 /etc/passwd
```

4.tail：显示文件尾部信息

重要选项

```
1 | -n
2 | -f
```

练习题：

显示文件最后10行文本

```
1 | tail /etc/passwd
```

显示文件最后5行文本

```
1 | tail -5 /etc/passwd
```

持续的显示文本的变化

```
1 | tail -f /var/log/message
```

5.less：分页显示文件内容

重要选项

less命令的快捷键与VIM快捷键类似

1	上下左右 或	上下翻页
2	-N	显示行号
3	q	退出
4	/关键词	搜索关键词
5	n	定位到下一个关键词
6	ctrl + u	往上翻页
7	ctrl + d	往下翻页

练习题

使用less命令查看/var/log/messages日志并显示行号并搜索eth0关键词

```
1 | less -N /var/log/messages
```

6.grep：文本过滤工具

功能说明

1. **grep**命令是Linux系统中最重要的命令之一，其功能是从文本或管道数据流中筛选匹配的行及数据。
2. 如果配合正则表达式技术一起使用，则功能会更加强大。
3. **grep**过滤就相当于一个筛子，有可能筛子里面的东西是要保留的，也有可能筛出来的需要保留。

重要选项

```
1 | -v  
2 | -n  
3 | -i  
4 | -E  
5 | -w
```

练习题：

过滤包含oldboy字符串的行

```
1 | grep "oldboy" /etc/passwd
```

过滤不包含oldboy字符串的行

```
1 | grep -v "oldboy" /etc/passwd
```

显示关键词所在行的行号

```
1 | grep -n oldboy /etc/passwd
```

显示所有行的行号

```
1 | grep -n . /etc/passwd
```

不区分大小写的过滤

```
1 | grep -ni "ipv6" ifcfg-eth0
```

过滤包含oldboy或oldya的行

```
1 | egrep -ni "ipv6|ipv4" ifcfg-eth0
```

精确搜索包含oldboy的字符串

```
1 | grep -w "oldboy" file1.txt
```

7.定向符号

功能说明

```
1 | ">"为重定向符号，表示清除原文件里的所有内容，然后将内容追加到文件的末尾(危险操作！慎用)
2 | ">>"为追加重定向符号，即追加内容到文件的尾部(文件的最后一行)
```

练习题:

清空一个文本

```
1 | > file1.txt
```

输出一行内容覆盖文本内容

```
1 | echo '123' > file1.txt
```

输出一行内容并追加到文本的最后

```
1 | echo '456' >> file1.txt
```

8.统计行号

```
1 | wc -l /var/log/nginx/access.log
```

