

# 目录

---

- 1 优化配置篇
  - 1.1 优化配置篇之打开文件显示行数
  - 1.1 优化配置篇之安装vim
  - 1.100 优化配置篇之总结脚本
- 2 Linux编程vi或vim高效率小技巧
- 3 Linux高效高频常用命令
  - 3.1 Linux高效高频之输入命令的常用命令
  - 3.2 Linux高效高频之文件或目录相关的常用命令
- 4 Linux中shell编程技巧与注意事项
  - 4.1 Linux中shell编程技巧
  - 4.2 Linux中shell编程注意事项

## 1.1 优化配置篇之打开文件显示行数

设置行号，每次打开都显示行号

修改vi ~/.vimrc 文件，添加：set number

## 1.2 优化配置篇之安装vim

sudo apt-get remove vim-common（该命令可以不执行）

sudo apt-get install vim

## 1.100 优化配置篇之总结脚本

```
#!/usr/bin/env bash

# 让文件显示行数
# sed -i '$a\set number' ~/.vimrc    #此方法只允许向已存在的文件末尾内输入
echo 'set number' >> ~/.vimrc

# 安装vim编辑器
sudo apt-get remove vim-common
sudo apt-get install vim
```

## 2 Linux编程vi或vim高效率小技巧

- 移动光标
  - h,j,k,l 上，下，左，右
  - ctrl-e 移动页面
  - ctrl-f 上翻一页
  - ctrl-b 下翻一页
  - ctrl-u 上翻半页
  - ctrl-d 下翻半页
  - w 跳到下一个字首，按标点或单词分割
  - W 跳到下一个字首，长跳，如end-of-line被认为是一个字
  - e 跳到下一个字尾
  - E 跳到下一个字尾，长跳

b 跳到上一个字  
B 跳到上一个字，长跳  
0 跳至行首，不管有无缩进，就是跳到第0个字符  
^ 跳至行首的第一个字符  
\$ 跳至行尾  
gg 跳至文首  
G 调至文尾  
5gg/5G 调至第5行

\* 查找光标所在处的单词，向下查找(\为反转义)  
# 查找光标所在处的单词，向上查找(\为反转义)

- 删除复制

dd 删除光标所在行  
dw 删除一个字(word)

- 插入模式

i 从当前光标处进入插入模式  
I 进入插入模式，并置光标于行首  
a 追加模式，置光标于当前光标之后  
A 追加模式，置光标于行末  
o 在当前行之下新加一行，并进入插入模式  
O 在当前行之上新加一行，并进入插入模式  
Esc 退出插入模式

- 退出编辑器

:w 将缓冲区写入文件，即保存修改  
:wq 保存修改并退出  
:x 保存修改并退出  
:q 退出，如果对缓冲区进行过修改，则会提示  
:q! 强制退出，放弃修改

- 查找替换

/pattern 向后搜索字符串pattern  
?pattern 向前搜索字符串pattern  
"lc" 忽略大小写  
"lC" 大小写敏感

n 下一个匹配(如果是/搜索，则是向下的下一个，?搜索则是向上的下一个)  
N 上一个匹配(同上)

:%s/old/new/g 搜索整个文件，将所有的old替换为new  
:%s/old/new/gc 搜索整个文件，将所有的old替换为new，每次都要你确认是否替换

### 3.1 Linux高效高频之输入命令的常用命令

- 移动相关

Ctrl + a-- 跳到行首  
Ctrl + e-- 跳到行尾

- 删除相关

Ctrl + d-- 向右删除一个字符  
Ctrl + h-- 向左删除一个字符  
Ctrl + u-- 删除当前位置字符至行首  
Ctrl + k-- 删除当前位置字符至行尾  
Ctrl + w -- 删除从光标到当前单词开头

- 其他

Ctrl + y-- 插入最近删除的单词  
Ctrl + c-- 终止操作  
Ctrl + d-- 当前操作转到后台  
Ctrl + l-- 清屏  
Ctrl + r-- 历史搜索

### 3.2 Linux高效高频之文件或目录相关的常用命令

cd -	回到上次操作的目录
du -sh	查看当前目录总共占的容量。而不单独列出各子项占用的容量
du -lh --max-depth=1	查看当前目录下一级子文件和子目录占用的磁盘容量
du -ah --max-depth=1	a表示显示目录下所有的文件和文件夹（不含子目录），h表示以人类能看懂的方式，max-depth表示目录的深度

## 4 Linux中shell编程技巧与注意事项

### 4.1 Linux中shell编程技巧

```SHELL  
技巧一：通过花括号定义函数的方式进行大区域注释，其他地方则不调用此函数

#### 参数

\$\$ Shell本身的PID（ProcessID）  
\$! Shell最后运行的后台Process的PID  
\$? 最后运行的命令的结束代码（返回值）  
\$- 使用Set命令设定的Flag一览  
\$\* 所有参数列表。如"\$\*"用「」括起来的情况、以"\$1 \$2 ... \$n"的形式输出所有参数。  
\$@ 所有参数列表。如"\$@"用「」括起来的情况、以"\$1" "\$2" ... "\$n" 的形式输出所有参数。  
\$# 添加到Shell的参数个数  
\$0 Shell本身的文件名  
```\n

### 4.2 Linux中shell编程注意事项

函数中的变量需要使用local定义为局部变量，否则全局会相互影响，shell中的变量默认是全局变量  
函数的调用，不应互相影响，设计时做好隔离性