



文本处理工具和正则表达式

讲师：王晓春

本章内容



马哥教育

IT 人的高薪职业学院

- ◆ 各种文本工具来查看、分析、统计文本
- ◆ grep
- ◆ 正则表达式
- ◆ 扩展正则表达式
- ◆ vim



抽取文本的工具

- ◆ 文件内容：less和 cat
- ◆ 文件截取：head和tail
- ◆ 按列抽取：cut
- ◆ 按关键字抽取：grep

◆ 文件查看命令：

cat, tac, rev

◆ cat [OPTION]... [FILE]...

-E: 显示行结束符\$

-n: 对显示出的每一行进行编号

-A: 显示所有控制符

-b: 非空行编号

-s: 压缩连续的空行成一行

◆ tac

◆ rev

分页查看文件内容

- ◆ more: 分页查看文件
more [OPTIONS...] FILE...
 - d: 显示翻页及退出提示
- ◆ less: 一页一页地查看文件或STDIN输出
查看时有用的命令包括:
 - /文本 搜索 文本
 - n/N 跳到下一个 或 上一个匹配
- less 命令是man命令使用的分页器

显示文本前或后行内容



- ◆ head [OPTION]... [FILE]...
 - c #: 指定获取前#字节
 - n #: 指定获取前#行
 - #: 指定行数
- ◆ tail [OPTION]... [FILE]...
 - c #: 指定获取后#字节
 - n #: 指定获取后#行
 - #: 同上
 - f: 跟踪显示文件fd新追加的内容,常用日志监控
相当于 --follow=descriptor
 - F: 跟踪文件名, 相当于--follow=name --retry
- ◆ tailf 类似tail -f, 当文件不增长时并不访问文件

按列抽取文本cut和合并文件paste

◆ cut [OPTION]... [FILE]...

-d DELIMITER: 指明分隔符, 默认tab

-f FILEDS:

#: 第#个字段

#,#[,#]: 离散的多个字段, 例如1,3,6

#-#: 连续的多个字段, 例如1-6

混合使用: 1-3,7

-c 按字符切割

--output-delimiter=STRING指定输出分隔符

cut和paste



◆ 显示文件或STDIN数据的指定列

```
cut -d: -f1 /etc/passwd
```

```
cat /etc/passwd | cut -d: -f7
```

```
cut -c2-5 /usr/share/dict/words
```

◆ paste 合并两个文件同行号的列到一行

```
paste [OPTION]... [FILE]...
```

-d 分隔符：指定分隔符，默认用TAB

-s：所有行合成一行显示

示例：

```
paste f1 f2
```

```
paste -s f1 f2
```


分析文本的工具

- ◆ 文本数据统计：wc
- ◆ 整理文本：sort
- ◆ 比较文件：diff和patch

收集文本统计数据wc



- ◆ 计数单词总数、行总数、字节总数和字符总数
- ◆ 可以对文件或STDIN中的数据运行

```
wc story.txt
```

```
39   237  1901 story.txt
```

行数 字数 字节数

◆ 常用选项

- -l 只计数行数
- -w 只计数单词总数
- -c 只计数字节总数
- -m 只计数字符总数
- -L 显示文件中最长行的长度

文本排序sort

- ◆ 把整理过的文本显示在STDOUT, 不改变原始文件

sort [options] file(s)

- ◆ 常用选项

- -r 执行反方向（由上至下）整理
- -R 随机排序
- -n 执行按数字大小整理
- -f 选项忽略（fold）字符串中的字符大小写
- -u 选项（独特，unique）删除输出中的重复行
- -t c 选项使用c做为字段界定符
- -k X 选项按照使用c字符分隔的X列来整理能够使用多次

◆ uniq命令：从输入中删除前后相接的重复的行

◆ uniq [OPTION]... [FILE]...

-c: 显示每行重复出现的次数

-d: 仅显示重复过的行

-u: 仅显示不曾重复的行

注：连续且完全相同方为重复

◆ 常和sort 命令一起配合使用：

```
sort userlist.txt | uniq -c
```

比较文件

◆ 比较两个文件之间的区别

```
diff foo.conf foo2.conf
```

```
5c5
```

```
< use_widgets = no
```

```
---
```

```
> use_widgets = yes
```

➤ 注明第5行有区别（改变）

复制对文件改变patch

- ◆ diff 命令的输出被保存在一种叫做“补丁”的文件中
 - 使用 -u 选项来输出“统一的 (unified)” diff 格式文件，最适用于补丁文件
 - ◆ patch 复制在其它文件中进行的改变（要谨慎使用）
 - 适用 -b 选项来自动备份改变了的文件
- ```
diff -u foo.conf foo2.conf > foo.patch
patch -b foo.conf foo.patch
```

- ◆ 1、找出ifconfig “网卡名” 命令结果中本机的IPv4地址
- ◆ 2、查出分区空间使用率的最大百分比值
- ◆ 3、查出用户UID最大值的用户名、UID及shell类型
- ◆ 4、查出/tmp的权限，以数字方式显示
- ◆ 5、统计当前连接本机的每个远程主机IP的连接数，并按从大到小排序

# Linux文本处理三剑客



马哥教育

IT 人的高薪职业学院

- ◆ grep: 文本过滤(模式: pattern)工具  
grep, egrep, fgrep (不支持正则表达式搜索)
- ◆ sed: stream editor, 文本编辑工具
- ◆ awk: Linux上的实现gawk, 文本报告生成器



## ◆ grep: Global search REgular expression and Print out the line

作用：文本搜索工具，根据用户指定的“模式”对目标文本逐行进行匹配检查；打印匹配到的行

模式：由正则表达式字符及文本字符所编写的过滤条件

## ◆ grep [OPTIONS] PATTERN [FILE...]

```
grep root /etc/passwd
```

```
grep "$USER" /etc/passwd
```

```
grep '$USER' /etc/passwd
```

```
grep `whoami` /etc/passwd
```

# grep命令选项



- ◆ --color=auto: 对匹配到的文本着色显示
- ◆ -v: 显示不被pattern匹配到的行
- ◆ -i: 忽略字符大小写
- ◆ -n: 显示匹配的行号
- ◆ -c: 统计匹配的行数
- ◆ -o: 仅显示匹配到的字符串
- ◆ -q: 静默模式, 不输出任何信息
- ◆ -A #: after, 后#行
- ◆ -B #: before, 前#行
- ◆ -C #: context, 前后各#行
- ◆ -e: 实现多个选项间的逻辑or关系  
    grep -e 'cat' -e 'dog' file
- ◆ -w: 匹配整个单词
- ◆ -E: 使用ERE
- ◆ -F: 相当于fgrep, 不支持正则表达式
- ◆ -f file: 根据模式文件处理

- ◆ REGEXP: Regular Expressions, 由一类特殊字符及文本字符所编写的模式, 其中有些字符 (元字符) 不表示字符字面意义, 而表示控制或通配的功能
- ◆ 程序支持: grep,sed,awk,vim, less,nginx,varnish等
- ◆ 分两类:
  - 基本正则表达式: BRE
  - 扩展正则表达式: ERE
  - grep -E, egrep
- ◆ 正则表达式引擎:
  - 采用不同算法, 检查处理正则表达式的软件模块
  - PCRE (Perl Compatible Regular Expressions)
- ◆ 元字符分类: 字符匹配、匹配次数、位置锚定、分组
- ◆ man 7 regex

# 基本正则表达式元字符



## ◆ 字符匹配:

- . 匹配任意单个字符

- [] 匹配指定范围内的任意单个字符, 示例: [wang] [0-9] [a-z] [a-zA-Z]

- [^] 匹配指定范围外的任意单个字符

- [:alnum:] 字母和数字

- [:alpha:] 代表任何英文大小写字符, 亦即 A-Z, a-z

- [:lower:] 小写字母 [:upper:] 大写字母

- [:blank:] 空白字符 (空格和制表符)

- [:space:] 水平和垂直的空白字符 (比[:blank:]包含的范围广)

- [:cntrl:] 不可打印的控制字符 (退格、删除、警铃...)

- [:digit:] 十进制数字 [:xdigit:] 十六进制数字

- [:graph:] 可打印的非空白字符

- [:print:] 可打印字符

- [:punct:] 标点符号

◆ 匹配次数：用在要指定次数的字符后面，用于指定前面的字符要出现的次数

\* 匹配前面的字符任意次，包括0次

贪婪模式：尽可能长的匹配

. \* 任意长度的任意字符

\? 匹配其前面的字符0或1次

\+ 匹配其前面的字符至少1次

\{n\} 匹配前面的字符n次

\{m,n\} 匹配前面的字符至少m次，至多n次

\{,n\} 匹配前面的字符至多n次

\{n,\} 匹配前面的字符至少n次

## ◆ 位置锚定：定位出现的位置

^ 行首锚定，用于模式的最左侧

\$ 行尾锚定，用于模式的最右侧

^PATTERN\$ 用于模式匹配整行

^\$ 空行

^[[:space:]]\*\$ 空白行

\< 或 \b 词首锚定，用于单词模式的左侧

\> 或 \b 词尾锚定，用于单词模式的右侧

\<PATTERN\> 匹配整个单词

- ◆ 分组：\(\) 将一个或多个字符捆绑在一起，当作一个整体处理，如：\(\root\)\+
- ◆ 分组括号中的模式匹配到的内容会被正则表达式引擎记录于内部的变量中，这些变量的命名方式为：\1, \2, \3, ...
- ◆ \1 表示从左侧起第一个左括号以及与之匹配右括号之间的模式所匹配到的字符
- ◆ 示例： \(\string1\+\(\string2\)\*\)  
    \1 : string1\+\(\string2\)\*  
    \2 : string2
- ◆ 后向引用：引用前面的分组括号中的模式所匹配字符，而非模式本身
- ◆ 或者：\|  
示例：a\|b: a或b   C\|cat: C或cat   \(\C\|c\)at: Cat或cat

# 正则表达式

| 元字符     | 定义                |
|---------|-------------------|
| ^       | 行首                |
| \$      | 行尾                |
| .       | 任意单一字符            |
| []      | []内任意单一字符         |
| [^]     | 除[]内任意单一字符        |
| *       | *前面字符重复不确定次数      |
| \+      | \+前面字符重复一次以上不确定次数 |
| \?      | ? 前面字符重复0或1次      |
| \       | 转义符               |
| .*      | 任意长度字符            |
| \{n\}   | 前面字符重复n次          |
| \{n,\}  | 前面字符重复n次以上        |
| \{m,n\} | 前面字符重复m次和n次之间     |



# 正则表达式

| 元字符       | 定义                            |
|-----------|-------------------------------|
| [alnum:]  | 字母和数字                         |
| [alpha:]  | 代表任何英文大小写字符，亦即 A-Z, a-z       |
| [lower:]  | 小写字母                          |
| [upper:]  | 大写字母                          |
| [blank:]  | 水平空白字符（空格和制表符）                |
| [space:]  | 所有水平和垂直的空白字符（比[blank:]包含的范围广） |
| [cntrl:]  | 不可打印的控制字符（退格、删除、警铃...）        |
| [digit:]  | 十进制数字                         |
| [graph:]  | 可打印的非空白字符                     |
| [print:]  | 可打印字符                         |
| [punct:]  | 标点符号                          |
| [xdigit:] | 十六进制数字                        |
|           |                               |

- ◆ 1、显示/proc/meminfo文件中以大小s开头的行(要求：使用两种方法)
- ◆ 2、显示/etc/passwd文件中不以/bin/bash结尾的行
- ◆ 3、显示用户rpc默认的shell程序
- ◆ 4、找出/etc/passwd中的两位或三位数
- ◆ 5、显示CentOS7的/etc/grub2.cfg文件中，至少以一个空白字符开头的且后面有非空白字符的行
- ◆ 6、找出“netstat -tan”命令结果中以LISTEN后跟任意多个空白字符结尾的行
- ◆ 7、显示CentOS7上所有系统用户的用户名和UID
- ◆ 8、添加用户bash、testbash、basher、sh、nologin(其shell为/sbin/nologin),找出/etc/passwd用户名和shell同名的行
- ◆ 9、利用df和grep，取出磁盘各分区利用率，并从大到小排序

# egrep及扩展的正则表达式



- ◆ `egrep = grep -E`
- ◆ `egrep [OPTIONS] PATTERN [FILE...]`
- ◆ 扩展正则表达式的元字符：
- ◆ 字符匹配：
  - . 任意单个字符
  - [] 指定范围的字符
  - [^] 不在指定范围的字符

# 扩展正则表达式



## ◆ 次数匹配:

\* 匹配前面字符任意次

? 0或1次

+ 1次或多次

{m} 匹配m次

{m,n} 至少m, 至多n次

# 扩展正则表达式



## ◆ 位置锚定:

^ 行首

\$ 行尾

\<, \b 语首

\>, \b 语尾

## ◆ 分组:

()

后向引用: \1, \2, ...

## ◆ 或者:

a|b            a或b

C|cat           C或cat

(C|c)at        Cat或cat

- ◆ 1、显示三个用户root、mage、wang的UID和默认shell
- ◆ 2、找出/etc/rc.d/init.d/functions文件中行首为某单词(包括下划线)后面跟一个小括号的行
- ◆ 3、使用egrep取出/etc/rc.d/init.d/functions中其基名
- ◆ 4、使用egrep取出上面路径的目录名
- ◆ 5、统计last命令中以root登录的每个主机IP地址登录次数
- ◆ 6、利用扩展正则表达式分别表示0-9、10-99、100-199、200-249、250-255
- ◆ 7、显示ifconfig命令结果中所有IPv4地址
- ◆ 8、将此字符串：welcome to magedu linux 中的每个字符去重并排序，重复次数多的排到前面

# vim简介



- ◆ vi: Visual Interface, 文本编辑器
- ◆ 文本: ASCII, Unicode
- ◆ 文本编辑种类:
  - 行编辑器: sed
  - 全屏编辑器: nano, vi
  - vim - Vi Improved
- ◆ 其他编辑器:
  - gedit 一个简单的图形编辑器
  - gvim 一个Vim编辑器的图形版本





# 打开文件



## ◆ # vim [OPTION]... FILE...

+ # 打开文件后，让光标处于第#行的行首，+默认行尾

+ /PATTERN 打开文件后，直接让光标处于第一个被PATTERN匹配到的行的行首

-b file 二进制方式打开文件

-d file1 file2... 比较多个文件

-m file 只读打开文件

## ◆ ex file 或 vim -e 直接进入ex模式

## ◆ 如果该文件存在，文件被打开并显示内容

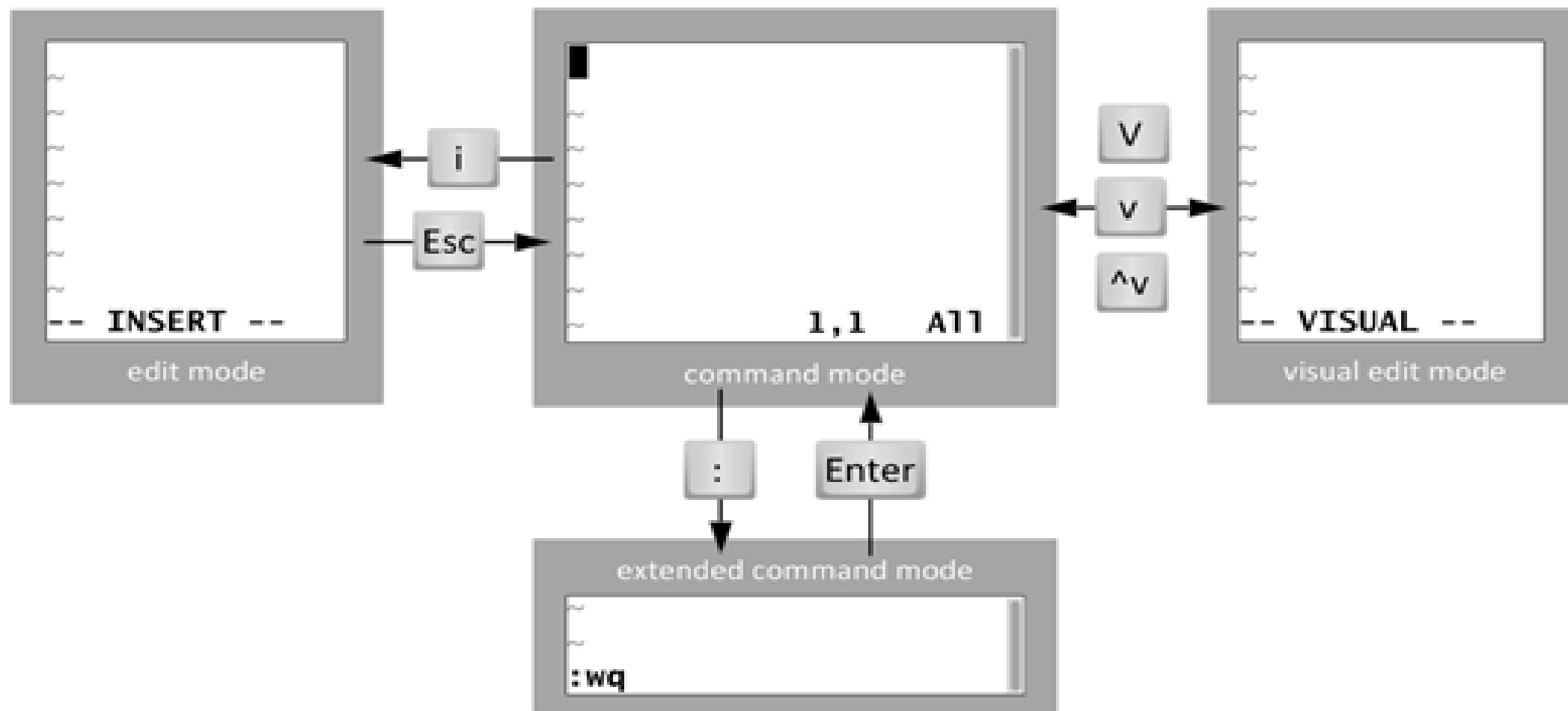
如果该文件不存在，当编辑后第一次存盘时创建它

# vim：一个模式编辑器



- ◆ 击键行为是依赖于 vim 的 “模式”
- ◆ 三种主要模式：
  - 命令(Normal)模式：默认模式，移动光标，剪切/粘贴文本
  - 插入(Insert)或编辑模式：修改文本
  - 扩展命令(extended command )模式：保存，退出等
- ◆ Esc键 退出当前模式
- ◆ Esc键 Esc键 总是返回到命令模式

# vim模式



## ◆ 命令模式 --> 插入模式

- i insert, 在光标所在处输入
- I 在当前光标所在行的行首输入
- a append, 在光标所在处后面输入
- A 在当前光标所在行的行尾输入
- o 在当前光标所在行的下方打开一个新行
- O 在当前光标所在行的上方打开一个新行

# 模式转换



◆ 插入模式 -----> 命令模式

ESC

◆ 命令模式 -----> 扩展命令模式

:

◆ 扩展命令模式 -----> 命令模式

ESC,enter

# 关闭文件



## ◆ 扩展命令模式：

:q 退出

:q! 强制退出，丢弃做出的修改

:wq 保存退出

:x 保存退出

## ◆ 命令模式

ZZ 保存退出

ZQ 不保存退出

# 扩展命令模式



- ◆ 按 “:” 进入Ex模式
- ◆ 创建一个命令提示符：处于底部的屏幕左侧
- ◆ 命令：
  - w      写（存）磁盘文件
  - wq     写入并退出
  - x      写入并退出
  - q      退出
  - q!     不存盘退出，即使更改都将丢失
  - r filename   读文件内容到当前文件中
  - w filename 将当前文件内容写入另一个文件
  - !command 执行命令
  - r!command 读入命令的输出

## ◆ 字符间跳转：

h: 左 l: 右 j: 下 k: 上

#COMMAND: 跳转由#指定的个数的字符

## ◆ 单词间跳转：

w: 下一个单词的词首

e: 当前或下一单词的词尾

b: 当前或前一个单词的词首

#COMMAND: 由#指定一次跳转的单词数

## ◆ 当前页跳转：

H: 页首 M: 页中间行 L: 页底

zt: 将光标所在当前行移到屏幕顶端

zz: 将光标所在当前行移到屏幕中间

zb: 将光标所在当前行移到屏幕底端



# 命令模式光标跳转



## ◆ 行首行尾跳转：

^: 跳转至行首的第一个非空白字符

0: 跳转至行首

\$: 跳转至行尾

## ◆ 行间移动：

#G、 扩展命令模式下：# 跳转至由#指定行

G: 最后一行

1G, gg: 第一行

## ◆ 句间移动：

): 下一句      (: 上一句

## ◆ 段落间移动：

}: 下一段      {: 上一段

# 命令模式翻页操作



马哥教育

IT 人的高薪职业学院

- ◆ Ctrl+f: 向文件尾部翻一屏
- ◆ Ctrl+b: 向文件首部翻一屏
- ◆ Ctrl+d: 向文件尾部翻半屏
- ◆ Ctrl+u: 向文件首部翻半屏

## ◆ 字符编辑：

x: 删除光标处的字符

#x: 删除光标处起始的#个字符

xp: 交换光标所在处的字符及其后面字符的位置

~:转换大小写

J:删除当前行后的换行符

## ◆ 替换命令(r, replace)

r: 替换光标所在处的字符

R:切换到REPLACE模式

## ◆ 删除命令：

d: 删除命令，可结合光标跳转字符，实现范围删除

d\$: 删除到行尾

d^: 删除到非空行首

d0: 删除到行首

dw:

de:

db:

#COMMAND

## ◆ dd: 删除光标所在的行

#dd: 多行删除

## ◆ D: 从当前光标位置一直删除到行尾，等同于d\$

# 命令模式操作



## ◆ 复制命令(y, yank):

y: 复制, 行为相似于d命令

y\$

y0

y^

ye

yw

yb

#COMMAND

## ◆ yy: 复制行

#yy: 复制多行

## ◆ Y: 复制整行

## ◆ 粘贴命令(p, paste):

p: 缓冲区存的如果为整行, 则粘贴当前光标所在行的下方; 否则, 则粘贴至当前光标所在处的后面

P: 缓冲区存的如果为整行, 则粘贴当前光标所在行的上方; 否则, 则粘贴至当前光标所在处的前面

- ◆ 改变命令(c, change)

- c: 修改后切换成插入模式

- ◆ 命令模式 --> 插入模式

- c\$

- c^

- c0

- cb

- ce

- cw

- #COMMAND

- ◆ cc: 删除当前行并输入新内容，相当于S

- #cc

- ◆ C: 删除当前光标到行尾，并切换成插入模式

# 命令模式： 操作文本

|                 | Change (replace) | Delete (cut) | Yank (copy) |
|-----------------|------------------|--------------|-------------|
| Line            | cc               | dd           | yy          |
| Letter          | cl               | dl           | yl          |
| Word            | cw               | dw           | yw          |
| Sentence ahead  | c)               | d)           | y)          |
| Sentence behind | c(               | d(           | y(          |
| Paragraph above | c{               | d{           | y{          |
| Paragraph below | c}               | d}           | y}          |



- ◆ 100iwang [ESC] 粘贴 “wang” 100次
  - ◆ <start position> <command> <end position>
  - ◆ Command:
    - y 复制、d 删除、gU 变大写、gu 变小写
- 例如 0y\$ 命令意味着：
- 0 → 先到行头
  - y → 从这里开始拷贝
  - \$ → 拷贝到本行最后一个字符
  - ye 从当前位置拷贝到本单词的最后一个字符

- ◆ di" 光标在" "之间, 则删除" "之间的内容
- ◆ yi( 光标在()之间, 则复制()之间的内容
- ◆ vi[ 光标在[]之间, 则选中[]之间的内容
- ◆ dtx 删除字符直到遇见光标之后的第一个 x 字符
- ◆ ytx 复制字符直到遇见光标之后的第一个 x 字符

# 扩展命令模式：地址定界



## ◆ 地址定界

:start\_pos,end\_pos

# 具体第#行，例如2表示第2行

#,# 从左侧#表示起始行，到右侧#表示结尾行

#,+# 从左侧#表示的起始行，加上右侧#表示的行数

： 2,+3 表示2到5行

. 当前行

\$ 最后一行

.,\$-1 当前行到倒数第二行

% 全文, 相当于1,\$

# 扩展命令模式：地址定界

## ◆ /pat1/,/pat2/

从第一次被pat1模式匹配到的行开始，一直到第一次被pat2匹配到的行结束

#,/pat/

/pat/, \$

## ◆ 使用方式：后跟一个编辑命令

d

y

w file: 将范围内的行另存至指定文件中

r file: 在指定位置插入指定文件中的所有内容

## ◆ 查找

/PATTERN：从当前光标所在处向文件尾部查找

?PATTERN：从当前光标所在处向文件首部查找

n：与命令同方向

N：与命令反方向

# 扩展命令模式：查找并替换

## ◆ s: 在扩展模式下完成查找替换操作

格式：s/要查找的内容/替换为的内容/修饰符

要查找的内容：可使用模式

替换为的内容：不能使用模式，但可以使用\1, \2, ...等后向引用符号；还可以使用 "&" 引用前面查找时查找到的整个内容

修饰符：

i: 忽略大小写

g: 全局替换；默认情况下，每一行只替换第一次出现

gc: 全局替换，每次替换前询问

## ◆ 查找替换中的分隔符/可替换为其它字符，例如

s@/etc@/var@g

s#/boot#/#i

- ◆ 1、复制/etc/profile至/tmp/目录，用查找替换命令删除/tmp/profile文件中的行首的空白字符
- ◆ 2、复制/etc/rc.d/init.d/functions文件至/tmp目录，用查找替换命令为/tmp/functions的每行开头为空白字符的行的行首添加一个#号

# 命令模式：撤消更改



- ◆ u撤销最近的更改
- ◆ #u撤销之前多次更改
- ◆ U 撤消光标落在这行后所有此行的更改
- ◆ 按Ctrl - r重做最后的“撤消”更改
- ◆ . 重复前一个操作
- ◆ n.重复前一个操作n次



- ◆ 有26个命名寄存器和1个无命名寄存器，常存放不同的剪贴版内容，可以不同会话间共享
- ◆ 寄存器名称a, b,...,z,格式：“寄存器 放在数字和命令之间  
如：3"tyy 表示复制3行到t寄存器中  
"tp 表示将t寄存器内容粘贴
- ◆ 未指定，将使用无命名寄存器
- ◆ 有10个数字寄存器，用0, 1, ..., 9表示，0存放最近复制内容，1存放最近删除内容。当新的文本变更和删除时，1转存到2，2转存到3，以此类推。数字寄存器不能在不同会话间共享

# 标记和宏(macro)

- ◆ ma 将当前位置标记为a, 26个字母均可做标记, mb、mc 等等;
- ◆ 'a 跳转到a标记的位置; 实用的文档内标记方法, 文档中跳跃编辑时很有用
- ◆ qa 录制宏 a, a为宏的名称
- ◆ q 停止录制宏,
- ◆ @a 执行宏 a
- ◆ @@ 重新执行上次执行的宏

- ◆ 以二进制方式打开文件

`vim -b binaryfile`

- ◆ 扩展命令模式下，利用xxd命令转换为可读的十六进制

`:%!xxd`

- ◆ 编辑二进制文件

- ◆ 扩展命令模式下，利用xxd命令转换回二进制

`:%!xxd -r`

- ◆ 保存退出

- ◆ 允许选择的文本块
  - ✓ 面向字符
  - ✓ 面向行
  - ctrl-v 面向块
- ◆ 可视化键可用于与移动键结合使用：
  - w ) } 箭头等
- ◆ 突出显示的文字可被删除，复制，变更，过滤，搜索，替换等

# 多文件模式



◆ vim FILE1 FILE2 FILE3 ...

:next 下一个

:prev 前一个

:first 第一个

:last 最后一个

:wall 保存所有

:qall 退出所有

:wqall

# 使用多个“窗口”



## ◆ 多文件分割

`vim -o|-O FILE1 FILE2 ...`

-o: 水平分割

-O: 垂直分割

在窗口间切换: `Ctrl+w`, Arrow

## ◆ 单文件窗口分割:

`Ctrl+w,s`: split, 水平分割

`Ctrl+w,v`: vertical, 垂直分割

`ctrl+w,q`: 取消相邻窗口

`ctrl+w,o`: 取消全部窗口

: wqall 退出

# 定制vim的工作特性



## ◆ 配置文件：永久有效

全局：/etc/vimrc

个人：~/.vimrc

## ◆ 扩展模式：当前vim进程有效

### ◆ (1) 行号

显示：set number, 简写为set nu

取消显示：set nonumber, 简写为set nonu

### ◆ (2) 忽略字符的大小写

启用：set ic

不忽略：set noic

### ◆ (3) 自动缩进

启用：set ai

禁用：set noai

# 定制vim的工作特性

## ◆ (4) 智能缩进

启用: `smartindent` 简写 `set si`

禁用: `set nosi`

## ◆ (5) 高亮搜索

启用: `set hlsearch`

禁用: `set nohlsearch`

## ◆ (6) 语法高亮

启用: `syntax on`

禁用: `syntax off`

## ◆ (7) 显示Tab和换行符 `^I` 和 `$` 显示

启用: `set list`

禁用: `set nolist`



# 定制vim的工作特性

## ◆ (8) 文件格式

启用windows格式: `set fileformat=dos`

启用unix格式: `set fileformat=unix`

简写: `set ff=dos|unix`

## ◆ (9) 设置文本宽度

`set textwidth=65 (vim only)`

`set wrapmargin=15`

## ◆ (10) 设置光标所在行的标识线

启用: `set cursorline`, 简写 `cul`

禁用: `set no cursorline`

## ◆ (11) 复制保留格式

启用: `set paste`

禁用: `set nopaste`

# 了解更多



马哥教育

IT 人的高薪职业学院

## ◆ Set 帮助

➤ :help option-list

➤ :set or :set all

## ◆ vi/vim内置帮助

:help

:help *topic*

Use :q to exit help

## ◆ vimtutor

- ◆ 1、在vim中设置tab缩进为4个字符
- ◆ 2、复制/etc/rc.d/init.d/functions文件至/tmp目录，替换/tmp/functions文件中的/etc/sysconfig/init为/var/log
- ◆ 3、删除/tmp/functions文件中所有以#开头，且#后面至少有一个空白字符的行的行首的#号

# vi / vim 键盘图

Esc

命令  
模式

|                     |                |                     |              |                     |                     |          |                    |                   |                       |                         |                       |                       |
|---------------------|----------------|---------------------|--------------|---------------------|---------------------|----------|--------------------|-------------------|-----------------------|-------------------------|-----------------------|-----------------------|
| ~ 转换大小写             | ! 外部过滤器        | @ 运行宏               | # prev ident | \$ 行尾               | % 括号匹配              | ^ "软"行首  | & 重复:s             | * next ident      | ( 句首                  | ) 下一句首                  | "soft" bol down       | + 后一行行首               |
| \. 跳转到标注            | 1 <sup>2</sup> | 2                   | 3            | 4                   | 5                   | 6        | 7                  | 8                 | 9                     | 0 "硬"行首                 | - 前一行行首               | = 自动 <sup>3</sup> 格式化 |
| Q 切换至ex模式           | W 下一单词         | E 词尾                | R 替换模式       | T back 'till        | Y 拷贝行               | U 撤销行内命令 | I 到行首插入            | O 分段(前)           | P 粘贴(前)               | { 段首                    | }                     | 段尾                    |
| q 录制宏               | w 下一单词         | e 词尾                | r 替换字符       | t 'till             | y 拷贝 <sup>1,3</sup> | u 撤销命令   | i 插入模式             | o 分段(后)           | p 粘贴 <sup>1</sup> (后) | [ 杂项                    | ]                     | 杂项                    |
| A 在行尾附加             | S 删除行并插入       | D 删除至行尾             | F 行内字符反向查找   | G 文尾/行号             | H 屏幕顶行              | J 合并两行   | K 帮助               | L 屏幕底行            | :                     | ex 命令                   | " 寄存器 <sup>1</sup> 标识 | 行首/列                  |
| a 附加                | s 删除字符并插入      | d 删除 <sup>1,3</sup> | f 行内字符查找     | g 附加命令 <sup>6</sup> | h ←                 | j ↓      | k ↑                | l →               | ;                     | 重复 <sup>1</sup> u/T/f/F | ' 跳转到标注的行首            | \. 未用!                |
| Z 退出 <sup>4</sup>   | X 退格           | C 修改至行末             | V 可视行模式      | B 前一单词              | N 查找上一处             | M 屏幕中间行  | < 反缩进 <sup>3</sup> | > 缩进 <sup>3</sup> | ?                     | 向前搜索                    |                       |                       |
| Z 附加命令 <sup>5</sup> | X 删除(字符)       | c 修改 <sup>1,3</sup> | V 可视模式       | b 前一单词              | n 查找下一处             | m 设置标注   | , 反向               | .                 | 重复命令                  | /                       | 向后搜索                  |                       |

动作

移动光标, 或者定义操作的范围

命令

直接执行的命令,  
红色命令 进入编辑模式

操作

后面跟随表示操作范围的指令

extra

特殊功能,  
需要额外的输入

q

后跟字符参数

w,e,b命令

小写(b): quux(foo, bar, baz);

大写(B): quux(Foo, Bar, Baz);

主要ex命令:

:w (保存), :q (退出), :q! (不保存退出)

:ef (打开文件 f),

:%s/x/y/g ('y' 全局替换 'x'),

:h (帮助 in vim), :new (新建文件 in vim),

其它重要命令:

CTRL-R: 重复 (vim),

CTRL-F/-B: 上翻/下翻,

CTRL-E/-Y: 上滚/下滚,

CTRL-V: 块可视模式 (vim only)

可视模式:

漫游后对选中的区域执行操作 (vim only)

备注:

(1) 在 拷贝/粘贴/删除 命令前使用 "x (x=a..z,\*)

使用命令的寄存器('剪贴板')

(如: "ay\$ 拷贝剩余的行内容至寄存器 'a')

(2) 命令前添加数字

多遍重复操作

(e.g.: 2p, d2w, 5i, d4j)

(3) 重复本字符在光标所在行执行操作

(dd = 删除本行, >> = 行首缩进)

(4) ZZ 保存退出, ZQ 不保存退出

(5) zt: 移动光标所在行至屏幕顶端,

zb: 底端, zz: 中间

(6) gg: 文首 (vim only),

gf: 打开光标处的文件名 (vim only)



# Vim 命令小抄

original card by Laurent Gregoires  
redesign by brohan

## 基本移动

|                |                            |
|----------------|----------------------------|
| <b>h l j k</b> | 左/右 移动一个字符, 上/下 移动一行       |
| <b>b w</b>     | 向 左/右 移动一个单词               |
| <b>ge e</b>    | 向 左/右 移动一个单词 (光标在词尾)       |
| <b>{ }</b>     | 移动到 前一个/后一个 段落开头           |
| <b>( )</b>     | 移动到 前一个/后一个 句子开头           |
| <b>0 gm</b>    | 移动到 行首/行正中                 |
| <b>^ \$</b>    | 移动到行 第一个非空白字符/行尾           |
| <b>nG ngg</b>  | 移动到第 <i>n</i> 行, 默认到 尾行/首行 |
| <b>n%</b>      | 移动到文件的 <i>n</i> 百分比处       |
| <b>n </b>      | 移动到当前行的第 <i>n</i> 列        |
| <b>nH nL</b>   | 移动到窗口 顶端/底部 第 <i>n</i> 行   |
| <b>%</b>       | 匹配下一个项目 (括号、方括号等.....)     |
| <b>M</b>       | 移动到窗口的中间行                  |

## 编辑修改

|                            |                          |
|----------------------------|--------------------------|
| <b>i a</b>                 | 在光标 前/后 插入文本             |
| <b>I A</b>                 | 在 行首/行尾 插入文本             |
| <b>gl</b>                  | 在第一列插入文本                 |
| <b>o O</b>                 | 在光标的 下方/上方 插入新行          |
| <b>rc</b>                  | 把光标下的字符换成 <i>c</i>       |
| <b>grc</b>                 | 类似 <i>r</i> , 但替换的是虚拟字符  |
| <b>R</b>                   | 进入替换模式                   |
| <b>gR</b>                  | 类似 <i>R</i> , 但替换的是虚拟字符  |
| <b>cm</b>                  | 更改动作 <i>m</i> 跨越的文本      |
| <b>CC or S</b>             | 更改当前行的内容                 |
| <b>C</b>                   | 更改光标到行尾的文本               |
| <b>s</b>                   | 更改一个字符并进入插入模式            |
| <b>~</b>                   | 切换当前字符大小写, 同时光标前移        |
| <b>g~m</b>                 | 切换动作 <i>m</i> 跨越文本的大小写   |
| <b>gum gUm</b>             | 小写/大写 动作 <i>m</i> 跨越的文本  |
| <b>&lt;m &gt;m</b>         | 左移/右移 动作 <i>m</i> 跨越的行   |
| <b>n&lt;&lt; n&gt;&gt;</b> | 将 <i>n</i> 行 左移/右移 一个缩进位 |

## 删除

|             |                                 |
|-------------|---------------------------------|
| <b>x X</b>  | 删除光标 后/前 的字符                    |
| <b>dm</b>   | 删除动作 <i>m</i> 跨越的文本             |
| <b>dd D</b> | 删除 当前行/到行尾                      |
| <b>J gJ</b> | 连接两行 删除缩进、插入空格/直接连接             |
| <b>rd</b>   | 删除范围 <i>r</i> 包括的行              |
| <b>rdx</b>  | 删除范围 <i>r</i> 包括的行到寄存器 <i>x</i> |

## 插入模式

|                       |                                 |
|-----------------------|---------------------------------|
| <b>^Vc ^Vn</b>        | 插入字符 <i>c</i> 的本义/十进制值 <i>n</i> |
| <b>^A</b>             | 插入最近插入的文本                       |
| <b>^@</b>             | 插入最近插入的文本, 并退出插入模式              |
| <b>^Rx</b>            | 原样插入寄存器 <i>x</i> 的内容            |
| <b>^N ^P</b>          | 补全关键字并作 正向/反向 查找                |
| <b>^W</b>             | 删除光标前的单词                        |
| <b>^U</b>             | 删除当前行光标前所有输入的字符                 |
| <b>^D ^T</b>          | 左移/右移 一个缩进位                     |
| <b>^KC1C2</b>         | 输入 <i>\C1,C2\</i> 代表的二合字母       |
| <b>^Oc</b>            | 在临时命令模式执行命令 <i>c</i>            |
| <b>^X^E</b>           | 向上滚动一行                          |
| <b>^X^Y</b>           | 向下滚动一行                          |
| <b>&lt;esc&gt; ^[</b> | 结束插入模式, 回到命令模式                  |

## 复制

|                |                            |
|----------------|----------------------------|
| <b>"x</b>      | 指定下次删除、抽出和放置使用寄存器 <i>x</i> |
| <b>:reg</b>    | 显示所有寄存器的内容                 |
| <b>:reg x</b>  | 显示寄存器 <i>x</i> 的内容         |
| <b>ym</b>      | 抽出动作 <i>m</i> 跨越的文本        |
| <b>yy or Y</b> | 抽出整行                       |
| <b>p P</b>     | 放置寄存器内容到光标 之后/之前           |
| <b>]p [P</b>   | 类似 "p" "P", 但调整当前行的缩进      |
| <b>gp gP</b>   | 类似 "p" "P", 光标停留在新文本之后     |

## 复杂改动

|                |                                     |
|----------------|-------------------------------------|
| <b>g?m</b>     | 对动作 <i>m</i> 跨越的文本做 rot13 编码        |
| <b>n^A n^X</b> | 光标之上或之后的数值或者字母 <i>+n/-n</i>         |
| <b>gqm</b>     | 格式化动作 <i>m</i> 跨越的所有行               |
| <b>:rce w</b>  | 将范围 <i>r</i> 中的行依据宽度 <i>w</i> 居中对齐  |
| <b>:rle i</b>  | 将范围 <i>r</i> 中的行靠左对齐, 缩进 <i>i</i> 列 |
| <b>:rri w</b>  | 将范围 <i>r</i> 中的行依据宽度 <i>w</i> 靠右对齐  |
| <b>lmo</b>     | 将动作 <i>m</i> 跨越的行用命令 <i>c</i> 过滤    |
| <b>n!c</b>     | 将 <i>n</i> 行用命令 <i>c</i> 过滤         |
| <b>rlc</b>     | 将范围 <i>r</i> 中的行用命令 <i>c</i> 过滤     |

## 可视模式

|                 |                             |
|-----------------|-----------------------------|
| <b>v V</b>      | 以 字符/行 方式开始 (结束) 高亮         |
| <b>^V</b>       | 以列块方式开始 (结束) 高亮             |
| <b>o</b>        | 交换高亮区域的开始处和光标位置             |
| <b>gv</b>       | 使用上一次的可视区域开始高亮              |
| <b>aw as ap</b> | 选择 "一个单词" / "一个句子" / "一个段落" |
| <b>ab aB</b>    | 选择 "一个块" ( ) / "一个大块" { }   |

## 撤消, 重做

|             |                     |
|-------------|---------------------|
| <b>u U</b>  | 撤销最近的改动 / 恢复最近被改动的行 |
| <b>. ^R</b> | 重复最近的改动 / 重做最近撤销的改动 |
| <b>n.</b>   | 重复最近的改动 <i>n</i> 次  |

## 寄存器

|                |                                                  |
|----------------|--------------------------------------------------|
| <b>qo qC</b>   | 记录键入的字符, 存入/添加到 寄存器 <i>c</i>                     |
| <b>q</b>       | 停止记录                                             |
| <b>@c</b>      | 执行寄存器 <i>c</i> 的内容                               |
| <b>@@</b>      | 重复上次的 @c 操作                                      |
| <b>:@c</b>     | 将寄存器 <i>c</i> 的内容当作 Ex 命令来执行                     |
| <b>:rg/p/c</b> | 在范围 <i>r</i> 中找到匹配内容 <i>p</i> 时执行 Ex 命令 <i>c</i> |

## 复合移动

|               |                                 |
|---------------|---------------------------------|
| <b>- +</b>    | 上移/下移 一行, 至第一个非空白字符             |
| <b>B W</b>    | 向 前/后 移动一个单词                    |
| <b>gE E</b>   | 向 前/后 移动一个单词 (光标在末尾)            |
| <b>n_</b>     | 下移 <i>n-1</i> 行, 至第一个非空白字符      |
| <b>g0</b>     | 移动到屏幕行第一个字符                     |
| <b>g^ g\$</b> | 移动到屏幕行 第一个非空白字符/最后一个字符          |
| <b>gk gj</b>  | 上移/下移 一个屏幕行                     |
| <b>fc Fc</b>  | 移动到 后一个/前一个 字符 <i>c</i> 处       |
| <b>tc Tc</b>  | 移动到 后一个/前一个 字符 <i>c</i> 之前      |
| <b>;;</b>     | 正向/反向 重复上次 "f"、"F"、"t" 或 "T" 命令 |
| <b>[[ ]]</b>  | 向后/向前 一小节, 置于小节的开始              |
| <b>[] ]]</b>  | 向后/向前 一小节, 置于小节的末尾              |
| <b>[( )]</b>  | 向后/向前 至未闭合的 '(' / ')'           |
| <b>[{ }]</b>  | 向后/向前 至未闭合的 '{' / '}'           |
| <b>[m ]m</b>  | 向后/向前 至 java method 的开始         |
| <b>[# ]#</b>  | 向后/向前 至未闭合的 #if, #else, #endif  |
| <b>[* ]*</b>  | 向后/向前 至注释的 开始/结束 /* */          |

## 查找替换

|                  |                                                                                      |
|------------------|--------------------------------------------------------------------------------------|
| <b>/s ?s</b>     | 向前/向后 查找 <i>s</i>                                                                    |
| <b>/s/o ?s/o</b> | 向前/向后 查找 <i>s</i> , 光标偏移量 <i>o</i>                                                   |
| <b>n or /</b>    | 向前重复上次查找                                                                             |
| <b>N or ?</b>    | 向后重复上次查找                                                                             |
| <b># *</b>       | 向后/向前 查找光标下的标识符                                                                      |
| <b>g# g*</b>     | 同 "#"/"*", 但也查找部分匹配                                                                  |
| <b>gd gD</b>     | 跳转到光标下标识符的 局部/全局 声明                                                                  |
| <b>:rs/f/t/x</b> | 将范围 <i>r</i> 中的 <i>f</i> 替换成 <i>t</i>                                                |
| <b>:rs x</b>     | 选项 <i>x</i> : <i>g</i> -所有匹配项, <i>c</i> -确认替换<br>在新的范围 <i>r</i> 中以选项 <i>x</i> 重复上一替换 |



# vim

## 复制/粘贴/删除

- yy复制当前行
- 5yy复制从当前行开始的5行
- p粘贴到下一行
- P粘贴到上一行
- D当前光标开始删除到行尾
- dd/.删除当前行
- ndd从当前行开始向后删除n行
- d1G删除第1行到当前行的数据
- dnG删除第n行到当前行的数据
- dG删除当前行到最后一行的数据
- x向后删除1个字符
- nx向后删除n个字符
- X向前删除1个字符

## 搜索

- /str从光标位置向下搜索str
- ?str从光标位置向上搜索str
- n找下一个
- N找上一个

## 替换

- :%s/abc/123/g把abc全部替换成123
- :%s/abc/123/gc同上，但要用户一个个确认是否替换

## 撤销

- u退回前一个动作
- ctrl+r重复前一个动作
- :e!文档还原到最原始状态

## 块操作

- v字符选择。光标经过的字符被选中
- V行选择。光标经过的行被选中
- ctrl+v块选择。矩阵方式选中
- y复制被选中的数据
- d删除被选中的数据

## 多文件编辑

- vim file1 file2同时编辑两个文件file1和file2
- :files列出vim当前打开的文件列表
- :n编辑下一个文件
- :N编辑上一个文件

## 多窗口操作

- :sp file2在打开一个文件后，用:sp在新窗口中打开另一个文件file2
- ctrl+w---->j窗口向下切换。先按ctrl+w，放开，再按j
- ctrl+w---->k窗口向上切换。先按ctrl+w，放开，再按k

## 其他操作

- set nu显示行号
- set nonu取消行号
- :r filename把filename文件里的内容插入到当前光标所在位子下
- :!command暂时离开vim，执行command命令。如:ldf -h

## 插入

- i在光标前插入
- I在行首插入
- a在光标后插入
- A在行末插入
- o在下一行插入
- O在上一行插入
- r输入替换光标位置字符
- R进入输入替换模式
- 大小写转换
- ESC退出编辑模式

## 保存

- :w保存
- :w!强制保存
- :w fileName文件另存为fileName
- n1,n2 w fileName将n1行到n2行的数据另存为fileName

## 退出

- :q退出
- :q!强制不保存退出
- :qw!强制保存退出
- :x保存并退出

## 光标移动

- 左下上右键
- h j k
- \$光标到行尾
- 0光标到行首
- G光标到文档最后一行
- gg光标到文档第一行
- nG/:n<enter>光标到文档第n行
- n<enter>向下移动n行
- n<space>向后移动n字符
- 向上翻页pageup
- ctrl+b
- 向下翻页pagedown
- ctrl+f

# 关于马哥教育



马哥教育

IT 人的高薪职业学院

- ◆ 博客: <http://mageedu.blog.51cto.com>
- ◆ 主页: <http://www.magedu.com>
- ◆ QQ: 1661815153, 113228115
- ◆ QQ群: 203585050, 279599283



# 祝大家学业有成

# 谢 谢

咨询热线 400-080-6560