第2章
文本文件处理

主要内容

2.1文本文件及处理工具

Linux中的文本信息

- **文本文件**
  - C语言，Java语言等编程文件的源程序语言
  - 文本格式的数据文件
  - 文本格式的文字信息
- **程序输出**
- **系统配置信息**
  - /etc下的配置文件（功能类似windows的注册表）
- **文本型网络协议因特网大部分传输层以上的协议是文本型协议**
  - 会话层协议：HTTP，POP3，SMTP，IMAP，
  - 表示层协议：HTML，XML，MIME
- **文本文件处理的命令**
  - Linux提供大量的文本文件处理的命令
  - 多数命令都有复杂的选项

进程的标准输入/输出，重定向，管道

- **进程的基本概念**
  - 进程与程序
  - 进程的生命周期
- **进程的输入输出**
  - 标准输入stdin
  - 标准输出stdout
- **重定向机制**
  - 输出重定向

  ls –l > filelist.txt
  - 输入重定向

  sort < filelist.txt
- **管道机制**

  ls –l | sort
- **重定向机制和管道机制的重要性**

文本文件处理命令的特点

- **特点**
  - 不指定处理对象文件名时，从标准输入获得数据
  - 指定处理对象文件名时，从文件中获取数据
  - 多数命令可以指定多个文件
  - 处理结果在标准输出显示
- **考虑的因素**
  - 标准输入/标准输出
  - shell的文件通配符
  - 输入输出重定向
  - 管道
- **灵活性：工具命令的组合**
  - Linux倾向于提供独立的多个精巧的工具命令，数据格式为文本信息
  - 鼓励使用重定向或管道机制将多个工具命令组合在一起，提供灵活的功能
  - 应用系统设计时，也应该考虑到这些特点
    - 例如：数据库内容的展示，直接输出多列文本，考虑到各种工具软件的使用

2.2 读取文件内容

介绍几个文本文件读取与处理的命令

- more/less：逐屏显示文件
- cat与od：列出文件内容
- head与tail：显示文件的头部或者尾部
- tee：三通
- wc：字计数
- sort：对文件内容排序
- tr：翻译字符
- uniq：筛选文件中的重复行

more/less：逐屏显示文件

- 历史
  - more：最先由BSD UNIX开发
  - less：Linux上广泛使用
- 使用方法
  - more shudu.c 指定一个文件
  - more *.[ch] 指定多个文件
  - ls -l | more 指定0个文件
  - less shudu.c

more

满屏后，显示--more--或--more--(15%)，可以使用more命令：

空格	显示下一屏
回车	上滚一行，当所感兴趣的段落内容正好处于当前屏的 尾部，另有一部分在下一页中时，可以连续按回车， 将感兴趣的部分滚动上来
q	(quit)退出程序，后面的内容不再显示
/*pattern*	搜索指定模式的字符串，模式描述用正则表达式
/	继续查找指定模式的字符串
h	(Help)帮助信息。打印more命令的所有功能列表
Ctrl-L	屏幕刷新

less

- **Linux系统中的命令less**

  less is more

  - 回退浏览的功能更强
  - 可直接使键盘的上下箭头键，或者j,k，类似vi的光标定位键，以及PgUp，PgDn，Home，End键
  - 有的Unix系统不提供less命令，但是可利用more命令的增强功能

cat与od：列出文件内容

- **基本功能与命名**
  - cat　concatenate:串结，文本格式打印 （选项-n：行号）
  - od　octal dump逐字节打印（-c, -t c, -t x1，-t d1, -t u1选项）
- **举例**
  - cat tryl.c　命令行参数1个
  - cat –n shudu.c
  - cat tryl.c tryx.c try.h 命令行参数3个
  - cat >try　命令行参数=0个，从stdin获取数据，直到ctrl-d
  - cat tryl.c try2.c try.h > trysrc
  - cat makefile *.[ch] > src

  - od –t x1 x.dat 以十六进制打印文件x.dat各字节
  - od –t x1 x.dat | more 以十六进制打印文件x.dat各字节
  - od –c /bin/bash　逐字符方式打印文件，遇到不可打印字符打印编码
  - echo abcdABCD | od –t x1 十六进制显示字符的ASCII码

head与tail：显示文件的头部或者尾部

默认只选择10行，-n选项可以选择行数

- head **–n** 15 ab.c

  显示文件ab.c中前15行

- head –n 23 a.c b.c c.c | more

  显示三个文件各自的前23行共显示69行

- tail –n 40　liu.mail
- head –n -20 msg.c 除去文件尾部20行其余均算“头”
- tail –n +20 msg.c 除去文件头部20行其余均算作“尾”
- tail -f debug.txt　实时打印文件尾部被追加的内容（选项-f:forever）

netstat -s -p tcp | head

ls -s | sort | head –n 20

tee：三通

- 功能
  - 将从标准输入stdin得到的数据抄送到标准输出stdout显示，同时存入磁盘文件中
- 应用举例
  - ./myap | tee myap.log

wc：字计数（word count）

- 功能
  - 列出文件中一共有多少行，有多少个单词，多少字符
  - 当指定的文件数大于1时，最后还列出一个合计
  - 常用选项-l：只列出行计数
- 举例
  - wc sum.c　　（1个文件）
  - wc x.c makefile stat.sh （多个文件）
  - wc -l *.c makefile start.sh
  - ps -ef | wc -l （0个）
  - ps -ef | grep liang | wc -l （0个）
  - who | wc -l （0个）

sort：对文件内容排序

- **sort选项**
  - -n选项(Numberic):对于数字按照算术值大小排序，而不是按照字符串比较规则，例如123与67
  - 可以选择行中某一部分作为排序关键字
  - 选择升序或降序
  - 字符串比较时对字母是否区分大小写
  - 内排序外排序等算法参数选择（当数据量较大时，性能调优）
- **举例**
  - sort telno > telno1
  - ls -s | sort | tail –10 →按字符
  - ls -s | sort -n | tail –10→按数字

tr：翻译字符

- **用法**

  `tr` *string1 string2*

  把标准输入拷贝到标准输出，*string1*中出现的字符替换为*string2*中的对应字符

- **例**

  ```
  cat telnos | tr UVX uvx
  ```

- **例：用［］指定一个集合**

  ```
  cat report | tr '[a-z]' '[A-Z]'
  ```

  将小写字母改为大写字母

- **例：用\加三个八进制数字(类似C语言)表示一字符**

  ```
  cat file1 | tr % '\012'
  ```

  将%改为换行符，注意不要漏掉必需的单引号

uniq：筛选文件中的重复行

- **用法**

  uniq ***options***

  uniq ***options input-file***

  uniq ***options input-file output-file***

- **重复的行：紧邻的两行内容相同**

```
Linux
Windows
Windows
Linux
Linux
Linux
AIX
```

- **选项**

  -u （uniqe）只保留没有重复的行

  -d （duplicated）只保留有重复的行（但只打印一次）

  没有以上两个选项，打印没有重复的行和有重复的行（但只打印一次)

  -c （count）计数同样的行出现几次

2.3 正则表达式及应用

正则表达式的概念

正则表达式的功能

- **正则表达式Regular Expressions应用范围**
  - 字符串匹配操作和替换操作
  - 举例：Linux中的vi more grep yacc lex awk sed
  - 其他：Visual Studio，Word等文本编辑器
- **正则表达式的功能**
  - 描述一个字符串模式
- **注意**
  - 正则表达式规则与文件名通配符规则不同
    - 正则表达式规则用于文本处理的场合
    - 文件名匹配规则用于文件处理的场合
  - 不同软件对正则表达式的定义会有差异

单字符正则表达式

- 长的正则表达式由单字符正则表达式构成的
- 非特殊字符与其自身匹配
  - 如：正则表达式a与字符串a匹配，b与b，/与/
- 转义字符（\）
  - \. * \$ \^ \[\\

正则表达式*与字符串*匹配，与字符串*不匹配

转义字符后除以上六种之外的不该出现其他字符，例如:不该出现\u，这样的组合被视为undefined（未定义的），后出的软件有可能会有特殊的解释

- 圆点（.）
  - 匹配任意单字符

单字符正则表达式：定义集合

- **基本用法**
  - 在一对方括号之间的字符为集合的内容，
    - 如：单字符正则表达式**[abcd]**与**a**或**b,c,d**匹配
  - 圆点,星号，反斜线在方括号内时，代表它们自己
    - 如**:[*.]**可匹配**3**个单字符
- **用减号-定义一个区间**
  - 如[a-d] [A-Z] [a-zA-Z0-9]
  - [][] 集合含左右中括号两个字符
  - 减号在最后，则失去表示区间的意义
    - **[ad-]**只与**3**个字符匹配
- **用^表示补集**
  - ^在开头,则表示与集合内字符之外的任意字符匹配
    - 如：**[^a-z]**匹配任一非小写字母
    - **[^][]**匹配任一非中括号字符
  - ^不在开头,则失去其表示补集的意义
    - 如：**[a-z^]**能匹配**27**个单字符

单字符正则表达式的组合(1)

- **串结**
  - 如abc， [A-Z].[0-9]·
- **星号（*）**
  - 单字符正则表达式后跟*，匹配此单字符正则表达式的**0次**或**任意多次**出现
- **例:正则表达式**12*4
  - 与字符串1234不匹配，与1224，12224，14匹配
- **例: 正则表达式**[A-Z][0-9]*

此例中*作用的单字符正则表式为[0-9]，代表

[A-Z]
[A-Z][0-9]
[A-Z][0-9][0-9]
[A-Z][0-9][0-9][0-9]，等等

与A,A1,C45,D768匹配，与b64512,T56t不匹配

单字符正则表达式的组合(2)

- **例：正则表达式**[Cc]hapter *[1-4]
  - 在*号前有一个空格，允许数字1-4之前有多个或者0个空格。可匹配Chapter2, chapter 3等等。
- **例：正则表达式**a\[i] *= *b\[j] ** *c\[k]
  - 匹配字符串a[i]=b[j]*c[k]，容许等号和星号两侧有空格
- **例：在vi中使用**

:1,$s/[0-9]*/xx/g

锚点：$与^

- **$ 在尾部时有特殊意义，否则与其自身匹配**
  - 例：123$ 匹配文件中行尾的123，不在行尾的123字符不匹配
  - 例：$123与字符串$123匹配
  - 例：.$ 匹配行尾的任意字符
- **^ 在首部时有特殊意义，否则与其自身匹配**
  - 例：^printf匹配行首的printf字符串，不在行首的printf串不匹配
  - 例：Hel^lo与字符串Hel^lo匹配
  - 例：在vi中使用 :10,50s/^ //g
    删除10-50行的每行行首的4个空格

正则表达式扩展

ERE：扩展的正则表达式(ERE)

PCRE: Perl-compatible regular expression

- **对基本正则表达式（BRE）进行了改进：**
  - 表示**分组**：圆括号()
  - 表示逻辑运算：表示逻辑“**或**” 的符号 |

    (xy)* 可匹配空字符串，**xy**，**xyxy**，**xyxyxy**

    (pink|green) 与**pink**或**green**匹配
  - **重复次数**定义：与星号地位类似的+和?，限定重复次数 \{m,n\}
    - ***号表示它左边的单字符正则表达式的0次或多次重复
    - **+**号表示**1**次或多次： **[0-9]+** 匹配长度至少为1数字串
    - **?**表示**0**次或一次： **a?** 匹配零个或一个**a**
    - 限定重复次数 **\{m,n\}**，例如：**[1-9][0-9]\{6,8\}** 7-9位数字，首位非0
  - 命名的预定义**集合**

    [[:xdigit:]] 十六进制数字

    \d数字 **\D**非数字
  - 比^和$更灵活的**锚点**定义

例如：寻找一个数字串，但是要求这个数字串不许出现在“合计”两个字之后

文本行筛选

grep在文件中查找字符串

grep（Global regular expression print ）

- **语法**

  grep 模式 文件名列表

- **举例**
  - grep O_RDWR *.h
  - ps -ef | grep liang
  - ls -l / | grep '^d' | wc –l
  - grep '[0-9]*' shudu.c
  - grep '[0-9][0-9]*' shudu.c

grep/egrep/fgrep

- **egrep 使用扩展正则表达式ERE描述模式**
  - 在指定模式方面比grep更灵活
- **fgrep 快速搜索指定字符串**
  - 按字符串搜索而不是按模式搜索。
- **grep选项**

-F, --fixed-strings Fixed strings (instead of regular expressions)

-G, --basic-regexp Basic regular expression (BRE)

-E, --extended-regexp Extended regular expression (ERE)

-P, --perl-regexp Perl-compatible regular expression (PCRE)

查PCRE语法：**man pcresyntax**

grep/fgrep/egrep选项

- 选项
  - -n 显示时每行前面显示行号
  - -v 显示所有不包含模式的行
  - -i 字母比较时忽略字母的大小写
- 例：grep -n main *.c
  - 查找含有正则表达式main的行，并打印行号
  - 当文件数超过一个时，除了输出行号，还输出文件名
- 例：grep -v '[Dd]isable' dev.stat>dev.active
  - 取消文件中所有含有指定模式的行，生成新文件
- 例：grep -i richard telnos
  - 在文件中检索字符串richard，不顾字母的大小写

流编辑及正则表达式替换

sed：流编辑

- **用法**
  - sed '命令' 文件名列表
  - sed **–e** '命令1' **–e** '命令2' **–e** '命令3' 文件名列表
  - sed **-f** 命令文件　文件名列表

  举例：
  - tail -f pppd.log | sed '**s**/145\.37\.23\.26/桥西/**g**'
  - tail -f pppd.log | sed -f sed.cmd
    - 其中 **sed.cmd** 文件

    s/145\.37\.23\.26/桥西**/g**

    s/102\.157\.23\.109/柳荫街**/g**

    s/145\.37\.123\.57/大山子**/g**
  - cat pm25.txt | sed 's/\[[^][]*]//g'

sed:正则表达式替换

- **模式描述中增加\(和\)**

  在正则表达式中圆括号，仍然代表它自身,在正则表达式中出现的**\(**和**\)**不影响匹配操作

  [a-zA-Z_][a-zA-Z0-9_]*->number

  \([a-zA-Z_][a-zA-Z0-9_]***\)**->number

- **替换字符串中的 \0 \1 \2 ……**

- **举例**

  - **s**/\([a-zA-Z_][a-zA-Z0-9_]*\)->number/\1->num/g
  - 将日期格式“月-日-年”改为“年.月.日”

    比如：将 04-26-1997替换为1997.04.26使用命令:

    s/\([0-9][0-9]\)-\([0-9][0-9]\)-\([0-9][0-9]*\)/\3.\1.\2/g
  - 生成文件改名的命令

  [快视频www.kuai-vdo.com]-电视剧《三国演义》中文字幕_1080p高清_央视1994版-69.rmvb

  sed 's/.*-**\([0-9][0-9]\)**.rmvb/mv \0 第**\1**集.rmvb/'

复杂筛选及加工awk

awk：逐行扫描进行文本处理

逐行扫描进行文本处理的一门语言，a.w.k分别为该程序的三位设计者姓氏的第一个字母

- 用法
  - awk '程序' 文件名列表
  - awk -f 程序文件名　文件名列表

  程序　**条件{动作}**

  一段程序：awk自动对每行文本执行**条件**判断，满足条件执行**动作**　（内置循环）

  多段程序：允许多段程序，程序间用空格或分号隔开

- 处理方式
  - **(行)**输入文件的每行作为一个**“记录”**，变量NR就是行号
  - **(列)**每行用空格分隔开的部分，叫做记录的**“域”**

    内置变量$1是第1域内容，依次，$2是第2域内容，……

    特别的，$0指的是整个这一行的内容
  - awk的处理为：符合条件的行，执行相应的动作

awk描述的条件

- 使用与C语言类似的**算数运算（加减乘除）**
- 使用与C语言类似的**关系运算**

  < 小于　　<= 小于或等于　　== 等于

  != 不等于　　> 大于　　>= 大于或等于
- 使用与C语言类似的**逻辑运算**

  || 条件或　　&& 条件与　　! 条件非
- 正则表达式的**模式匹配**
  - */regexpr/* 包含该模式的行
  - 正则表达式匹配运算符　~　　!~
    - 例如：　$2 ~ "[1-9][0-9]*"
- 特殊的条件
  - 不指定任何条件，对所有文本行执行动作
  - BEGIN　开始处理所有文本行之前执行动作
  - END　　处理完所有文本行之后执行动作

awk描述的动作

- 描述“动作”时，简单的用法有：
  - 自定义变量的赋值
  - 流程控制（与C语言类似）
    - 条件判断 `if`
    - 循环控制 `for`
    - 表示程序块的`{}`
  - `print 变量1, 变量2, ……`
  - `printf(”格式串”, 变量1, 变量2, ……)`

awk举例

```
$ ps -ef | grep guest
   guest   669   668  0 11:27:13   ttyp1    00:00:00 -sh
   guest   678   669  0 11:27:18   ttyp1    00:00:00 vi
$ ps -ef | awk '/guest/{ printf("%s ",$2); }‘
669 678
$ awk '{printf("%d: %s\n",NR,$0); }' test.c
1: main()
2: {
3:     printf("Hello!\n");
4: }
$ date
Thu May 27 22:02:22 BEIDT 2004
$ date | awk '{print $4}'
22:02:42
$ ls -s | awk '$1 > 2000 { print $2 }'
disk.img
document.pdf
linux-src.tar.Z
pppd.log
```

正则表达式应用

例1 累加作品发行量

- 1997年 于《少年文艺》等刊物发表《生涯模式》等作品。
- 1999年 获得“首届全国新概念作文比赛”一等奖 。初赛作品《求医》和《书店》，复赛作品《杯中窥人》。随后因期末考试七科不及格而留级，引发社会关于“学校应当培养全才还是专才”等系列教育问题的激烈讨论。
- 2000年 凭借作品《穿着棉袄洗澡》获得“第二届全国新概念作文比赛”二等奖。
- 2000年 出版长篇小说《三重门》畅销200 多万册，成为中国近二十年来销量最大的文学类作品。
- 2001年 出版文集（散文、随笔和短篇小说的合集）《零下一度》，畅销110多万册，2001年全国图书畅销排行榜第一。
- 2002年 出版小说《像少年啦飞驰》畅销100多万册，该书还出了同名漫画
- 2002年 出版作品精选集《毒》、《毒2》畅销80多万册
- 2003年 出版杂文集《通稿2003》畅销90多万册，2003年全国文学类畅销图书排名第三。
- 2004年 出版小说《长安乱》畅销160多万册，2004年全国图书排行榜文学类畅销书第一。
- 2004年 出版文集《韩寒五年文集》畅销78多万册。作品有法国、韩国、香港、新加坡、台湾、日本版本。其中法国版本获得法国2004年10月法国最畅销图书。

例2 绘制游客流量随时间变化的曲线

获取网页的命令：wget http://s.visitbeijing.com.cn/flow.php -o /dev/null; mv flow.php 180801.html

2018-08-01 北京景区舒适度指数播报(9点至16点每小时更新一次数据)

景区\时间	9：00		10:00		11:00		12:00		13:00		14:00		15:00		16:00		
	人数(万人)	舒适度	人数(万人)	舒适度	人数(万人)	舒适度	人数(万人)	舒适度	人数(万人)	舒适度	人数(万人)	舒适度	人数(万人)	舒适度	人数(万人)	舒适度	路况
故宫	约1.26	4	约2.01	4	约2.51	4	约2.45	4	约2.23	4	约1.82	4	约1.55	4	约1.29	4	查看
天坛公园	约1.06	5	约1.12	5	约1.09	5	约1.03	5	约0.96	5	约0.97	5	约1.00	5	约1.00	5	查看
恭王府	约0.07	5	约0.07	5	约0.08	5	约0.08	5	约0.08	5	约0.08	5	约0.09	5	约0.09	5	查看
国家体育场	暂无	暂无	暂无	暂无	暂无	暂无	暂无	暂无	暂无	暂无	暂无	暂无	暂无	暂无	暂无	暂无	查看
国家游泳中心	暂无	暂无	暂无	暂无	暂无	暂无	暂无	暂无	暂无	暂无	暂无	暂无	暂无	暂无	暂无	暂无	查看
奥林匹克森林公园	约4.72	5	约4.85	5	约5.13	4	约5.30	4	约5.49	4	约5.80	4	约5.97	4	约6.05	3	查看
颐和园	约1.26	5	约1.43	5	约1.56	5	约1.59	5	约1.61	5	约1.62	5	约1.61	5	约1.54	5	查看
八达岭	约0.93	4	约0.96	4	约0.78	4	约0.77	4	约0.72	4	约0.64	4	约0.60	5	约0.46	5	查看
十三陵	约0.89	5	约0.96	5	约1.04	5	约1.04	5	约1.05	5	约0.99	5	约0.95	5	约0.90	5	查看
慕田峪长城	约0.16	5	约0.21	5	约0.27	5	约0.28	5	约0.28	5	约0.25	5	约0.20	5	约0.16	5	查看
龙潭湖公园	约0.49	5	约0.50	5	约0.47	5	约0.43	5	约0.42	5	约0.44	5	约0.42	5	约0.41	5	查看
中山公园	约0.41	5	约0.41	5	约0.44	5	约0.35	5	约0.24	5	约0.23	5	约0.21	5	约0.19	5	查看

例2 绘制游客流量随时间变化的曲线

期望得到的输出结果（组织成.csv文件）：

故宫,2018-08-01 9:00,1.26

故宫,2018-08-01 10:00,2.01

故宫,2018-08-01 11:00,2.51

故宫,2018-08-01 12:00,2.45

故宫,2018-08-01 13:00,2.23

故宫,2018-08-01 14:00,1.82

故宫,2018-08-01 15:00,1.55

故宫,2018-08-01 16:00,1.29

天坛公园,2018-08-01 9:00,1.06

天坛公园,2018-08-01 10:00,1.12

天坛公园,2018-08-01 11:00,1.09

例3 观察TCP发送窗口变化

文件upload.cap是在Windows上使用软件WireShark捕获的文件上载过程中的网络流量。根据捕获的网络流量观察TCP发送窗口变化

执行命令tcpdump -ttnr upload.cap

```
1228623354.154058 IP 192.168.0.108.2301 > 123.117.178.136.20: Flags
[S.], seq 3210618050, ack 4098564053, win 65535, options [mss
1460,nop,wscale 0,nop,nop,TS val 0 ecr 0,nop,nop,sackOK], length 0

1228623354.424039 IP 123.117.178.136.20 > 192.168.0.108.2301: Flags
[.], ack 10081, win 6528, options [nop,nop,TS val 1620443362 ecr
129980], length 0

1228623354.424059 IP 192.168.0.108.2301 > 123.117.178.136.20: Flags
[.], seq 20161:21601, ack 1, win 65535, options [nop,nop,TS val
129981 ecr 1620443362], length 1440

1228623354.424094 IP 192.168.0.108.2301 > 123.117.178.136.20: Flags
[.], seq 21601:23041, ack 1, win 65535, options [nop,nop,TS val
129981 ecr 1620443362], length 1440
```

期望得到的结果:

```
20.835455,5760
20.868127,5760
20.868165,7200
20.902815,7200
20.902853,8640
20.933327,8640
20.933365,10080
20.965792,10080
20.965829,11520
20.998753,11520
20.998788,12960
21.031269,12960
21.031318,14400
21.063662,14400
21.063710,15840
```

作业1：正则表达式应用

从因特网上搜索相关Web网页，处理网页html数据，从中提取出当前时间点北京各监测站的PM2.5浓度，输出格式如下。要求：写出各个处理步骤，并给出解释。

2020-03-09 13:00:00,海淀区万柳,73
2020-03-09 13:00:00,昌平镇,67
2020-03-09 13:00:00,奥体中心,66
2020-03-09 13:00:00,海淀区万柳,73
2020-03-09 13:00:00,昌平镇,73
2020-03-09 13:00:00,奥体中心,75

北邮爱课堂平台提交

2.4 文件比对

文件内容比对

文件比较

- **基本功能**
  - cmp和散列算法：判断两个文件内容是否相同
  - 文件数据完整性验证
  - diff：列出两个文本文件之间的区别
  - 版本管理

两文件逐字节比较：cmp

- **用法**
  - cmp *file1 file2*
- **功能**
  - 逐字节比较两个文件是否完全相同
  - 两个文件完全相同时，不给出任何提示
  - 两个文件不同时，打印出第一个不同之处
  - 在Windows中有类似的命令**COMP**

md5sum/sha1sum:文件内容比较

- 使用MD5算法（散列函数）根据文件内容生成16字节hash值，比较hash值是否相同，就可断定两文件内容是否完全相同
- 使用SHA-1算法的命令名为**sha1sum** (20字节hash值)
  - 常用于数据完整性（**Data Integrity**）验证和判断位于网络不同机器上的两个文件内容是否相同
  - **其他散列函数也可以用来完成这一任务: sha512sum**

md5sum/sha1sum:文件内容比较

```
$ md5sum src.tar proto.txt
4faffc1f3714a693d7844dcb949ce020  src.tar
937d369a72c240a01a6a7a9efba919bd  proto.txt
2b00042f7481c7b056c4b410d28f33cf  log.txt
 $ md5sum src.tar proto.txt log.txt > myfiles.sum
```

仅将**myfiles.sum**传送到另台计算机，在另台计算机上运行程序，比较同名文件的内容是否一致

```
$ md5sum -c myfiles.sum
src.tar: OK
proto.txt: OK
log.txt: OK
```

失误率

MD5: $2^{-128} = 3.4 \times 10^{-38}$

SHA-1: $2^{-160} = 4.7 \times 10^{-50}$

求文本文件版本间差异

求出两个文件的差别：diff

- **用法**

  diff *file1 file2*

  diff –u *file1 file2*

- **功能**
  - 比较两个版本的文本文件，以寻找两者间差别
  - 输出格式 normal，unified **(-u)**
  - normal格式：列出一个如何将*file1*转化为*file2*的指令
    - 这些指令有**a**（**Add**），**c**（**Change**）和**d**（**Delete**）
    - 指令字母左边的行号是***file1***的行号，右面是***file2***的行号
    - 列出内容时，大于号后边的内容是需要在***file1***文件中增加的内容；小于号后边的内容是需从***file1***中删除的内容

normal格式文件转化指令

指令	如何将原文件转化为新文件
25**c**25,26 < #define MAX_WEIGHT 2000000000 --- > /* max signed 32-bit integer */ > #define MAX_WEIGHT 0x7fffffff	将原文件的第25行更换(change)成新文件的的第25~26行
61,62**d**60 < x2 = ht[j].weight; < m2 = j;	将原文件第6l~62行删除(delete)后，后面的内容与新文件的第60行之后一致
68**a**67,68 > ht[i].parent = -1; > printf("Create new tree\n");	将原文件的第68行增加(add)新文件中的第67~68行

cmp和diff举例

```
$ cmp ht.c ht2.c
ht.c ht2.c differ: byte 1074, line 25
$ diff  ht.c ht2.c
25c25,26
< #define MAX_WEIGHT 2000000000
---
> /* max signed 32-bit integer */
> #define MAX_WEIGHT 0x7fffffff
50,51c51
<         x1 = x2 = MAX_WEIGHT;
<         m1 = m2 = 0;
---
>         x1 = MAX_WEIGHT;
61,62d60
<                 x2 = ht[j].weight;
<                 m2 = j;
68a67,68
>         ht[i].parent = -1;
>         printf("Create new tree\n");
```

diff -u举例

```
$ diff -u0 ht.c ht2.c
--- ht.c        2018-11-14 18:18:59.595497610 +0800
+++ ht2.c       2018-11-17 08:22:58.191858259 +0800
@@ -25 +25,2 @@
-#define MAX_WEIGHT 2000000000
+/* max signed 32-bit integer */
+#define MAX_WEIGHT 0x7fffffff
@@ -50,2 +51 @@
-        x1 = x2 = MAX_WEIGHT;
-        m1 = m2 = 0;
+        x1 = MAX_WEIGHT;
@@ -61,2 +60,0 @@
-                x2 = ht[j].weight;
-                m2 = j;
@@ -68,0 +67,2 @@
+        ht[i].parent = -1;
+        printf("Create new tree\n");
```

版本管理系统

- **几种版本管理系统**
  - 1975 SCCS
  - 1986 CVS （Concurrent Versions System）
  - 2001 SVN
  - 2005 GIT
- **基本功能**
  - 对一个目录树(项目)下的文本文件(主要是源程序文件和脚本以及文档)进行版本管理
  - 便于多人合作开发的项目(一个代码库，多个现场)
  - 可以通过网络访问代码库

2.5 vi编辑器

编辑状态和光标移动

用户的偏好设置

- **用户HOME目录下的文件.exrc，记作$HOME/.exrc (每用户一份，用户独立设置)**

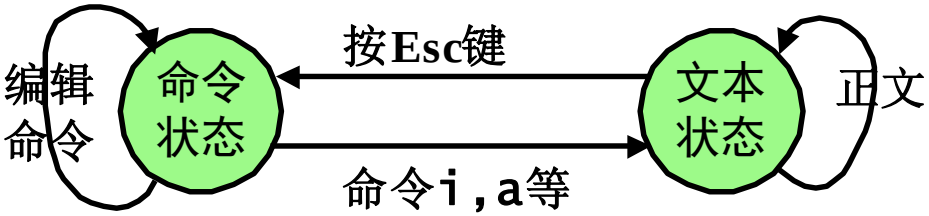
  set number　　每行左边显示行号

  set tabstop=4　制表符位置为4格对齐

- **用运行时检查偏好设置**

  :set

vi的两种工作状态



- **命令状态：键盘输入解释为命令**
  - vi一启动就进入命令方式，键盘输入解释为命令
  - 一般按键无回显
  - 以冒号可以引入行编辑的命令和查找命令
  - 编辑命令 i a 等，可以从命令状态转到文本状态
- **文本状态**
  - 键盘输入解释为输入的文本
  - 可以输入多行，每输入完一行后按回车转入下一行
  - 正文输入时有回显
  - 输入完毕按键盘左上角的Esc键，返回到命令状态
- **正文插入**
  - 命令 i 在当前字符前插入正文段，直至按Esc键(insert)
  - 命令 a 在当前字符后插入正文段，直至按Esc键(append)

光标单字符移动

- **单字符移动（四个字母键盘上相邻的按键）**
  - **h** 光标左移一列
  - **j** 光标下移一行
  - **k** 光标上移一行
  - **l** 光标右移一列
  - 一般可以直接使用键盘上的方向键代替这四个字母
- **命令前加一整数，表示这个命令连续执行多少遍**
  - **5h** 光标左移5列
  - **6j** 光标下移6行
  - **23k** 光标上移23行
  - **10l** 光标右移10列

注意：在vi命令状态下的按键命令没有回显

翻页

- **命令**
  - **Ctrl-b** 向后翻页(Backward)
  - **Ctrl-f** 向前翻页(Forward)
  - 一般可以用**PgDn**键代替Ctrl-f，用**PgUp**键代替Ctrl-b
  - 也可以使用下面的命令
    - **6Ctrl-f** 向前翻**6**页
    - **15Ctrl-b** 向后翻**15**页

光标行内快速移动

- **行尾行首**
  - 将光标移至当前行首 **^**
  - 将光标移至当前行尾 **$**
- **移动一个单词**
  - 移到右一个单词 **w**
  - 移到左一个单词 **b**
  - 也可以使用6w 5b 命令

光标移动到指定行

- **移到指定的行**
  - **:476**　将光标定位于第476行
  - **:1**　将光标定位于第1行（文件首）
  - **:$**　将光标定位于文件尾
- **在描述行号时可以使用**
  - 圆点（.）代表当前行号，
  - $　代表最后一行的行号
- **括号配对 %**
  - 把光标移到一个花括号(或圆括号，或方括号)上，按%键，则光标自动定位到与它配对的那一个括号

查找、编辑及存盘

删除命令

- **删除字符**
  - 删除当前字符的命令　**x**
  - 命令5x删除从当前光标开始的5个字符
- **删除行**
  - 删除当前行的命令　**dd**
  - 命令3dd删除从当前行开始的3行

字符替换

- **替换光标处字符 r**
  - ra命令将当前光标处字符替换为a
  - 将当前光标处开始的三个字符依次替换为abc，则需要按命令rarbrc

取消和重复

- **取消上一次的编辑操作(undo) u**
  - 如：误删了一段正文，用u命令可撤销删除
  - 如：把文件中的所有abc字符串替换成xyz字符串， 用u命令可撤销替换
- **重复上一次的编辑操作 .**
  - 按圆点键，可以重复上一次的编辑操作
  - 例如：按3dd命令删除了三行，然后按圆点键就再删除三行，接着连续按圆点键，每按一次删三行

文件操作命令

- 存盘退出
  - ZZ
  - :wq<CR>
- 存盘不退出
  - :w<CR>
- 不存盘退出
  - :q!<CR>
- 读入文件xyz.c插入到当前行之下
  - :r xyz.c<CR>
- 写文件,把第50行至文件尾的内容写到文件file1中
  - :50,$w file1<CR>
  - :50,$w! file1<CR>　　强制覆盖

剪贴板

- 删除，并拷贝到剪贴板
  - **:10,50d**<CR>　　删除第10-50行
  - **:1,.d**<CR>　　删除文件首至当前行的部分
  - **:.,$d**<CR>　　删除当前行到文件尾
- 不删除，拷贝到剪贴板(yank)
  - **:10,50y**<CR>
- 粘贴剪贴板信息（paste）
  - **p**

块操作：复制与删除

- **复制**
  - **:5,10co56**<CR>　复制第5-10行到第56行之下
- **移动**
  - **:8,34m78**<CR>　移动第8-34行到第78行之下

行合并、刷屏和状态显示

- **两行合并(Join)　J**
  - 当前行下面的行合并到当前行
- **刷新屏幕显示(load)　Ctrl-l**
- **状态显示 Ctrl-g**
  - 在屏幕最下面一行列出正在编辑的文件的名字，总行数，当前行号，文件是否被修改过等信息

模式查找

用“正则表达式”来描述一个字符串模式

- **查找命令**
  - 格式　　/*pattern*
  - 例：　　/[0-9][0-9]*
- **继续查找命令**
  - **n** 向下查找下一个next
  - **N** 向上查找下一个
  - 循环式搜索（向下搜索时遇到文件尾则回到文件头继续搜索）

模式替换

- **替换命令 (substitution)**
  - 格式　　*:n1,n2*s/*pattern*/*string*/g
  - 例
    - :1,50s/abc/xyz/
    - :1,50s/abc/xyz/g
    - :50,80s/^/　　/　　第50-75行右移4列
    - :50,80s/^　　//　　第50-75行左移4列
    - :1,$s/ *$//　　消除尾部多余的空格
    - :1,$s/a[i]/b[j]/g　　小心陷阱：不能把a[i]替换为b[j]
    - :1,$/a*b/x+y/g　　小心陷阱：不能把a*b替换为x+y

模式替换中的转义符

尤其是编辑C语言源程序时需要

- **将a[i]*b[j]替换为x[k]*y[n]的命令**
  - :1,$s/a\[i]*b\[j]/x[k]*y[n]/g
- **将buf.len/1000替为buffer.size/1024的命令**
  - :1,$s/buf\.len\/1000/buffer.size\/1024/g

模式串和替换字符串中的斜线前加转义符\以区别于替换命令格式中所必须的斜线

- :1,$s:buf\.len/1000:buffer.size/1024:g

s后面以冒号取代斜线，分界符就换为冒号，避免对斜线的转义

:1,$s^http://www\.myvdo\.com/a/b/c/index\.html^https://www.xyvdo.com/index.html^g

更灵活的替换

- **模式描述中增加\(和\)**
  - 在正则表达式中圆括号，仍然代表它自身
  - 在正则表达式中出现的**\(**和**\)**不影响匹配操作
- **例**
  - [a-zA-Z_][a-zA-Z0-9_]*->number
  - **\(**[a-zA-Z_][a-zA-Z0-9_]***\)**->number
- **替换字符串中的 \0 \1 \2 ……**
- **将“变量名->number”替换为“变量名->num”**

  :1,$s/\([a-zA-Z_][a-zA-Z0-9_]*\)->number/\1->num/g
- **将日期格式“月-日-年”改为“年.月.日”，**

  比如：将 04-26-1997替换为1997.04.26使用命令:

  :1,$s/\([0-9][0-9]\)-\([0-9][0-9]\)-\([0-9][0-9]*\)/\3.\1.\2/g

所谓“死机”问题

- **现象**

  vi编辑结束后执行存盘操作，结果导致屏幕卡死，输入任何信息都不再有显示（死机，终端死机）

- **原因**

  vi编辑结束后按下Ctrl-S，因为Windows编辑器一般设置Ctrl-S热键的动作为Save，但Linux却进入流量控制状态

- **解决方法**

  按下Ctrl-Q键后流量控制解除

Ctrl-S

Ctrl-Q.

意外中止问题

- **现象**

  vi编辑结束后存盘，程序意外中止，编辑成果丢失，文件内容未发生变化

- **原因**

  vi存盘命令Shift-ZZ，误操作为Ctrl-ZZ，而Ctrl-Z按键导致当前运行进程被**挂起（suspend）**，暂停运行（但进程尚在，处于Stopped状态）

- **解决方法**

  调用bash的作业管理机制，恢复运行被Stopped的进程

  `jobs` 列表当前被Stopped的进程有哪些

  `fg %1` 将1号作业恢复到前台（foreground）运行

  `%1` 将1号作业恢复到前台（foreground）运行