文本处理工具和正则表达式

讲师：王晓春

本章内容

- 各种文本工具来查看、分析、统计文本
- 文本处理三剑客之grep
- 正则表达式
- 扩展正则表达式

抽取文本的工具

- 文件内容： cat ,more,less
- 文件截取：head,tail
- 按列抽取：cut
- 排序和统计：sort,wc
- 按关键字抽取：grep

文件查看

- 文件查看命令：

  cat，nl，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

- nl
- tac
- rev

查看非文本文件内容

- hexdump

```
hexdump  -C -n 512 /dev/sda
00000000  eb 63 90 10 8e d0 bc 00  b0 b8 00 00 8e d8 8e c0  |.c..............|
echo {a..z} | tr -d ' '|hexdump  -C
00000000  61 62 63 64 65 66 67 68  69 6a 6b 6c 6d 6e 6f 70  |abcdefghijklmnop|
00000010  71 72 73 74 75 76 77 78  79 7a 0a                 |qrstuvwxyz.|
0000001b
```

- od：dump files in octal and other formats

```
echo {a..z} | tr -d ' '|od -A x -t x1z
000000 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70  >abcdefghijklmnop<
000010 71 72 73 74 75 76 77 78 79 7a 0a                 >qrstuvwxyz.<
00001b
```

- xxd

```
echo {a..z} | tr -d ' '|xxd
0000000: 6162 6364 6566 6768 696a 6b6c 6d6e 6f70  abcdefghijklmnop
0000010: 7172 7374 7576 7778 797a 0a               qrstuvwxyz.
```

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
  - -#　同上
- 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

- 可用于统计文件的行总数、单词总数、字节总数和字符总数
- 可以对文件或STDIN中的数据统计

```
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 #　选项按照使用c字符分隔的 # 列来整理能够使用多次

uniq

- 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
```

练习

- 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

- 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

- 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
- -m # 匹配#次后停止
- -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

正则表达式

- 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：vim, less,grep,sed,awk, nginx,varnish 等
- 分两类：
  - 基本正则表达式：BRE，grep，vim
  - 扩展正则表达式：ERE，grep -E, egrep，nginx
- 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

正则表达式

- 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

正则表达式

- 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

正则表达式

- 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

练习

- 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UID小于1000以内的用户名和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

练习

- 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

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