

# 正则表达式

教师：李军辉

# grep命令

分析一行信息，如果当中有我们所需要的信息，就将该行拿出来。

**grep [-acinv] [-color=auto]** '查找字符串' filename

选项与参数：

- a：将 binary 文件以 text 文件的方式搜寻数据
- c：计算找到 '搜寻字符串' 的次数
- i：忽略大小写的不同，所以大小写视为相同
- n：顺便输出行号
- v：反向选择，亦即显示出没有 '搜寻字符串' 内容的那一行！
- color=auto**：可以将找到的关键词部分加上颜色的显示！

- (1) 将 last 当中，有出现 root 的那一行就取出来；
- (2) 与范例一相反，只要没有 root 的就取出。
- (3) 在 last 的输出信息中，只要有 root 就取出，并且仅取第一栏。
- (4) 取出 /etc/man.config 内含 MANPATH 的那几行

- (1) 将 last 当中，有出现 root 的那一行就取出来；
- (2) 与范例一相反，只要没有 root 的就取出。
- (3) 在 last 的输出信息中，只要有 root 就取出，并且仅取第一栏。
- (4) 取出 /etc/man.config 内含 MANPATH 的那几行

```
last | grep 'root'  
last | grep -v 'root'  
last | grep 'root' | cut -d ' ' -f1  
grep --color=auto 'MANPATH' /etc/man.config
```

# 更多grep使用说明

<http://www.cnblogs.com/peida/archive/2012/12/17/2821195.html>

# awk文本处理工具

- awk基本命令格式
- 常见用法示例

# 关于文本处理

awk 编程语言/数据处理引擎

- 发明者: Aho、Weinberger、Kernighan
- 基于模式匹配检查输入
- 将期望的匹配结果 print 到屏幕

# awk基本命令格式

语法格式：

- awk ‘模式 [操作]’ 文件1 文件2 .....

# awk基本命令格式

语法格式：

- awk ‘模式 [操作]’ 文件1 文件2 .....
- awk ‘NR==1 {print}’ /etc/hosts

# awk基本命令格式

## 常见的内置变量

- NR 当前处理行的序数 (行号)
- FS 字符分隔符 (默认为空格和Tab)
- \$n 当前行的第n个字段
- \$0 当前行的所有文本内容

# awk用法示例

file.txt

```
1 This is the first line .
2 Hello , Everybody !
3 192.168.1.1 www.suda.edu.cn
4 lijunhui:x:504:504::/home/lijunhui:/bin
```

# awk用法示例

按行号输出文本：

- awk 'NR==1, NR==3 {print}' file.txt

# awk用法示例

按行号输出文本：

- awk 'NR==1, NR==3 {print}' file.txt
- awk '(NR==1) || (NR==3) {print}' file.txt

# awk用法示例

按行号输出文本：

- awk 'NR==1, NR==3 {print}' file.txt
- awk '(NR==1) || (NR==3) {print}' file.txt
- awk '(NR%2)==1 {print}' file.txt

# awk用法示例

按行号输出文本：

- awk 'NR==1, NR==3 {print}' file.txt
- awk '(NR==1) || (NR==3) {print}' file.txt
- awk '(NR%2)==1 {print}' file.txt
- awk '(NR%2)==0 {print}' file.txt

# awk练习

last 可以获取登陆者的信息，“-n 5”表示取前5行

```
last -n 5
```

若我想要取出帐号与登陆者的 IP，且帐号与 IP 之间  
以 [tab] 隔开，怎么操作??

```
last -n 5 | awk '{print $1 "\t" $3}'
```

`last` 可以获取登陆者的信息, “`-n 5`”表示取前5行

`last -n 5`

- 列出每一行的帐号(就是 `$1`)；
- 列出目前处理的行数(就是 `awk` 内的 `NR` 变量)
- 并且说明，该行有多少栏位(就是 `awk` 内的 `NF` 变量)

`last` 可以获取登陆者的信息，“`-n 5`”表示取前5行

`last -n 5`

- 列出每一行的帐号(就是 `$1`)；
- 列出目前处理的行数(就是 `awk` 内的 `NR` 变量)
- 并且说明，该行有多少栏位(就是 `awk` 内的 `NF` 变量)

`last -n 5 | awk '{print $1 "\t lines: " NR "\t columns: " NF}'`

# awk用法示例

使用正则表达式：

- awk '/2/ {print}' file.txt

# awk用法示例

使用正则表达式：

- awk '/2/ {print}' file.txt
- awk '/bash\$/ {print}' file.txt

指定分隔、指定输出字段：

- awk ‘NR==2, NR==3 {print \$1, \$3}’ file.txt

指定分隔、指定输出字段：

- awk ‘NR==2, NR==3 {print \$1, \$3}’ file.txt
- awk -F ‘\$5==”suda” {print \$0}’ file.txt

## 条件运行符

- `>` : 大于
- `<` : 小于
- `>=` : 大于或等于
- `<=` : 小于或等于
- `==` : 等于
- `!=` : 不等于

在 /etc/passwd 当中是以冒号 ":" 来作为栏位的分隔，该文件中第一栏位为帐号，第三栏位则是 UID。那假设我要查阅，第三栏小于 10 以下的数据，并且仅列出帐号与第三栏。

在 /etc/passwd 当中是以冒号 ":" 来作为栏位的分隔，该文件中第一栏位为帐号，第三栏位则是 UID。那假设我要查阅，第三栏小于 10 以下的数据，并且仅列出帐号与第三栏。

```
cat /etc/passwd | awk '{FS=":"} $3 < 10 {print $1 "\t" $3}'
```

在 /etc/passwd 当中是以冒号 ":" 来作为栏位的分隔，该文件中第一栏位为帐号，第三栏位则是 UID。那假设我要查阅，第三栏小于 10 以下的数据，并且仅列出帐号与第三栏。

```
cat /etc/passwd | awk '{FS=":"} $3 < 10 {print $1 "\t" $3}'
```

```
cat /etc/passwd | awk 'BEGIN {FS=":"} $3 < 10 {print $1 "\t" $3}'
```

# awk的计算功能

pay.txt

Name	1st	2nd	3th
VBird	23000	24000	25000
DMTsai	21000	20000	23000
Bird2	43000	42000	41000

如何计算每个人的总额呢？而且还要格式化输出！

- 第一行只是说明，所以第一行不要进行加总 (NR==1 时处理)；
- 第二行以后就会有加总的情况出现 (NR>=2 以后处理).

```
cat pay.txt | awk 'NR==1{printf "%10s %10s %10s  
%10s %10s\n",$1,$2,$3,$4,"Total"} NR>=2{total =  
$2 + $3 + $4 printf "%10s %10d %10d %10d  
%10.2f\n", $1, $2, $3, $4, total}'
```

# sed文本处理工具

- sed基本用法
- 输出文本
- 删除及替换文本

# sed基本用法

Sed 流式编辑器 / 文本过滤

- Stream Editor
- 基于模式匹配修改和替换文本

# sed基本命令格式

语法格式：

- `sed '编辑指令' 文件1 文件2 .....`
- `sed -n '编辑指令' 文件1 文件2 .....`
- `sed -i '编辑指令' 文件1 文件2 .....`

# sed基本命令格式

语法格式：

- `sed ‘编辑指令’ 文件1 文件2 .....`
- `sed -n ‘编辑指令’ 文件1 文件2 .....`
- `sed -i ‘编辑指令’ 文件1 文件2 .....`
- `sed -n ‘3,5p’ /etc/hosts`

# sed基本命令格式

编辑指令的写法：

- 格式：[地址1 [,地址2]] 操作类型
- 多条指令之间用分号隔开

# sed基本命令格式

编辑指令的写法：

- 格式：[地址1 [,地址2]] 操作类型
- 多条指令之间用分号隔开
- `sed -n '3,5p' /etc/hosts`
- `sed -n '3p;5p' /etc/hosts`

# sed基本命令格式

操作类型：

- p    输出打印文本行
- n    取下一行文本(跳过当前行)
- d    删除
- s    替换
- a    追加新的文本

# sed用法示例

file.txt

```
1 This is the first line.  
2 Hello , Everybody !  
3 192.168.1.1 www.suda.edu.cn  
4 lij unhui:x:504:504::/home/lijunhui:/bin
```

# sed用法示例

隔行输出文本内容：

- `sed -n 'p;n' file.txt`
- `sed -n 'n;p' file.txt`

# sed用法示例

隔行输出文本内容：

- `sed -n 'p;n' file.txt`
- `sed -n 'n;p' file.txt`

-n选项：只显示我们要打印的行，无关紧要的内容不显示

# sed用法示例

使用正则表达式：

- `sed -n '/suda/,\$p' file.txt`
- `sed -n '/\<This\>/p' file.txt`

# sed用法示例

删除符合条件的行:

- `sed '2,3d' file.txt`
- `sed '/suda/,$/d' file.txt`

# sed用法示例

删除符合条件的行:

- `sed '2,3d' file.txt`
- `sed '/suda/d;$d' file.txt`

# sed用法示例

删除不符合条件的行:

- sed '/suda/!d' file.txt

# sed用法示例

替换符合条件的行:

- sed '3,4s/suda/nju/g' file.txt

# sed用法示例

替换的特殊效果：

- sed '3,4s/^#/‘ file.txt
- sed 's/suda//g' file.txt

# sed用法示例

替换的特殊效果(修改原文本内容):

- `sed -i '3,4s/^#/ ' file.txt`
- `sed -i 's/suda//g' file.txt`