Linux之父：找份Linux工作是明智选择

  
 1991年， 也就是25年前托瓦兹以个人兴趣爱好启动了Linux的开发， 他想要打造一款满足他需求的操作系统。时至今日， Linux俨然成为了数以千计GNU/Linux操作系统和项目的幕后功臣。  
 Linux已经涉及日常生活的方方面面， 驱动互联网的大型服务器、 汽车、 移动手机和部分嵌入式设备等都能够看到Linux的身影。 但是更为重要的是Linux并非都是枯燥的技术， 在托瓦兹看来背后的Linux社区涌现了很多富有创意的开源项目，为你的未来职业生涯创造了无限可能。 因此托瓦兹推荐你考虑选择在这片领域中发展， 这或许将会成为你这辈子做的最明智决定。 这个世界需要更多的Linux和开源专家。

Linux系统一切皆文件，对Linux的操作就是对文本的操作。针对文本操作，有“Linux文本操作三剑客”——grep、sed、awk。其中grep命令擅长查找功能；sed擅长取行和替换；awk擅长取列。这三款工具均可结合正则表达式，达到精确操纵文本，进而操纵Linux系统。

学习时采用精简原则--熟练掌握最基本最常见用法。

程序=数据结构+算法

学习一门编程语言，我们需要学习什么？

Shell 自动化流程处理，特色：能够直接调用linux命令、能够之间使用Linux机制，比如管道，重定向

语言-->语法：

变量、运算符（算术、逻辑）

逻辑控制语句：选择结构、循环结构、顺序结构。

函数概念

程序设计思想：面向过程、面向对象

程序生成方式：编译、解释

常见正则表达式实例学习，结合find、grep、后续的awk和sed

SED、AWK学习

SED通过管道（|）和AWK配合使用。

1. 几个常用的环境变量的使用，P110 表3-2

Bash环境变量添加。

1. 位置变量的使用，这个很重要，和后续脚本编程挂钩
2. Shell一些测试选项 P114 表3-4
3. 正则表达式使用，集合grep命令和find命令

-----------------------------------

awk、sed、grep可以结合管道、重定向，也可以写在脚本中。

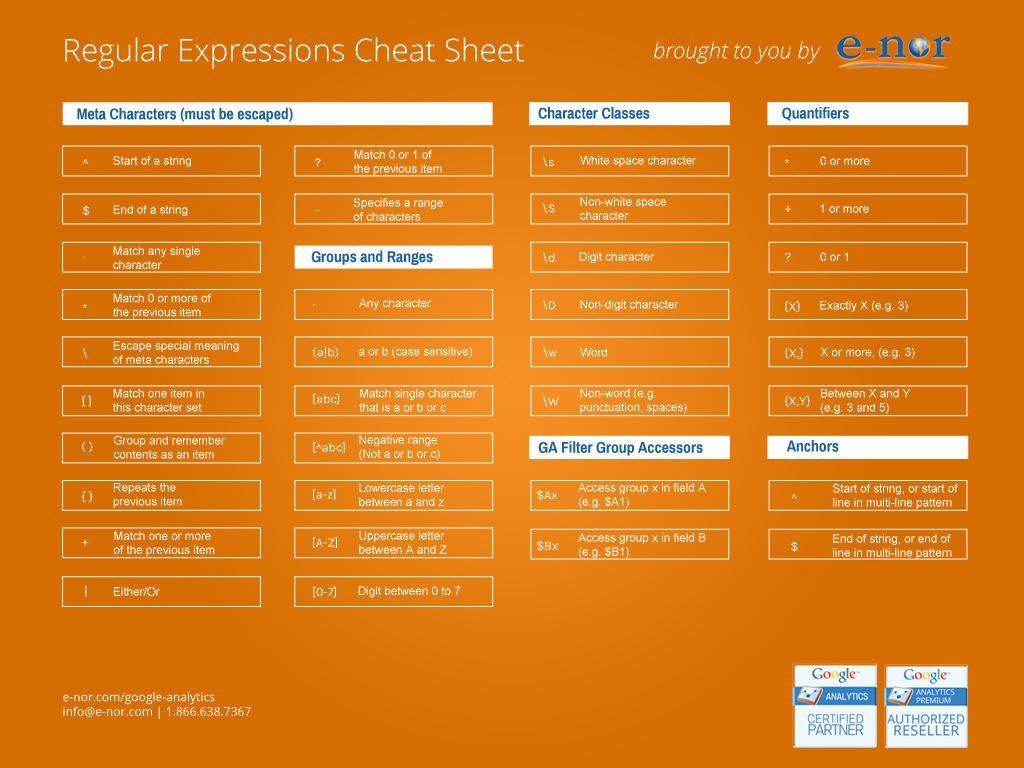
先生成示范文本

rpm -qai >> software-info

rpm -qai | head -15 >sedtest

### 一、正则表达式

正则表达式是组成“样式”的基本语法，而“样式”是运用 sed 和 awk 必备的能力。 sed和 awk 相同的运行方式是：只要符合“样式”的数据行，就对它执行指定的“操作”。



结合grep语法

grep -E -o “match\_pattern" file\_name1 file\_name2 file\_name3

-o 表示只输出文件中匹配到的部分

-E 表示使用扩展正则表达式

rpm -qai | head -15 > re-test

### 二、SED简明教程

sed语法如下：

sed '样式命令' 文件

sed 的工作方式是读取文件内容，经流编辑之后，把结果显示到标准输出。

sed默认是将结果输出到屏幕上，因此其不会改变文件内容，想要改变文件内容，需要加一个 -i 选项；或者重定向到临时文件，然后确认临时文件没问题后，用mv 将临时文件覆盖源文件。

#删除1行到5行数据，并修改到源文件

sed -i '1,5d' sedtest

#删除第3行数据，并修改到源文件

sed -i '3d' sedtest

也可也这么操作：

sed '3d' sedtest > tmp

cat tmp //查看tmp文件内容，确定没问题后，再替换

mv tmp sedtest

#将S打头的行，打印出来,-n选项意味着只输出符合条件的

sed -n '/^S/p' sedtest

#删除空白行

sed '/^$/d' sedtest

#将Bulid替换成Make，并打印输出到屏幕

sed -n 's/Build/Make/' sedtest

#将包含2016那一行中的Build替换为Make

sed -n '/2016/s/Build/Make/p' sedtest

#将第一行到第七行，所有的6替换成x，而不是第一个6

[root@localhost jkjk]# sed -n '1,7s/6/x/pg' sedtest

### 三、AWK简明教程

awk '样式'文件：把符合样式的数据行显示出来。

awk '{操作}' 文件：对每一行都执行{}中的操作。

awk '样式{操作}' 文件：对符合样式的数据行，执行{}中的操作。

#显示URL所在行

awk '/URL/' software\_info

#打印出以：为分割，第二列

awk -F : '{print $2 }' software\_info

cat software\_info | head -20 | awk -F: '{print $2}'

#查找含有La关键字的数据行，中第三，四列，默认列分隔符是空格

awk '/La/{ print $3,$4}' 文件

#将ftp打头的账户1~3栏信息打印出来，以+++作为输出分隔符，以：作为输入分割符

awk -F: 'BEGIN{OFS="++++"} /^ftp/{print $1,$2,$3}' /etc/passwd

##获取网卡IP(方法有很多)

1.ifconfig | grep 'broadcast' | awk '{print $2}'

2.ifconfig | grep inet | awk '{print $2}' | head -1

3.ifconfig | awk '/inet/{print $2}' | head -1

3.hostname -I

### 四、grep用法

1.作用

Linux系统中grep命令是一种强大的文本搜索工具，它能使用正则表达式搜索文本，并把匹配的行打印出来。grep全称是Global Regular Expression Print，表示全局正则表达式版本，它的使用权限是所有用户。

2.格式

grep [options] XXX文件

3.主要参数

[options]主要参数：

－c：只输出匹配行的计数。

－i：不区分大小写(只适用于单字符)。

－h：查询多文件时不显示文件名。

－l：查询多文件时只输出包含匹配字符的文件名。

－n：显示匹配行及行号。

－s：不显示不存在或无匹配文本的错误信息。

－v：显示不包含匹配文本的所有行。

pattern正则表达式主要参数：

操作符 含义

. 匹配任何单个字符。

? 匹配前一个字符0次或1次。

\* 匹配前一个字符≥0次。

+ 匹配前一个字符≥1次。

{N} 匹配前一个字符N次。

{N,} 匹配前一个字符≥N次。

{N,M} 匹配前一个字符 N 到 M次。

– 如果在列表中的某个列表或某个范围内的结束点，表示该范围。

^ 开始标记，表示在开始位置匹配一个空字符串。也表示不在列表的范围内的字符。

$ 结束标记。匹配一个空的字符串。

\b 单词锁定符。在一个单词的边缘位置匹配空字符串。

\B 在一个单词的非边缘位置匹配空字符串。

\< 匹配单词开始的空字符串。

\> 匹配单词结尾的空字符串。

[ ] 单个字符，如[A]即A符合要求 。

[ - ] 范围，如[A-Z]，即A、B、C一直到Z都符合要求 。

1. 常用实例

(1)多个文件查询

grep "sort" \*.doc #见文件名的匹配

1. 行匹配:输出匹配行的计数

grep -c "48" data.doc #输出文档中含有48字符的行数

(3)显示匹配行和行数

grep -n "48" data.doc #显示所有匹配48的行和行号

(4)显示非匹配的行

grep -vn "48" data.doc #输出所有不包含48的行

(5)大小写不敏感

grep -i "ab" data.doc #输出所有含有ab或Ab,AB,aB的字符串的行

5.结合正则表达式的应用

(1)匹配特定范围的

grep '[^239].' data.doc #输出所有含有以2,3或9开头的，并且是两个数字的行

(2)不匹配测试

grep '^[^48]' data.doc #不匹配行首是48的行

(3)使用扩展模式匹配

grep -F '219|216' data.doc

### 五、其他一些命令

sort，uniq

(1)uniq命令用于报告或忽略文件中的重复行，一般与sort命令结合使用。

sort file.txt | uniq

(2)sort,sort命令将文件的每一行作为一个单位，相互比较，比较原则是从首字符向后，依次按ASCII码值进行比较，最后将他们按升序输出。

相关选项：

-u : unique 唯一，排序并且排除重复项

-r : reverse 反向排序

-o : output 将排序结果输出到源文件

sort默认将结果输出到标准输出，所以需要重定向才可以将结果写入文件，形如 sort filename > newfile，但是如果你想把排序结果输出到原文件中，用重定向不可行

-n ： number 按照数字进行排序

-t ： tables 指定分隔符，默认tab为分隔符

-k： 指定安装哪一个分割区域进行排序

**sort -n -k 2 -t : facebook.txt**

# -n 按照数字进行排序

# -t 此例中指定 ： 为分隔符

# -k 此例中按照分割的第二区域进行排序

其他的sort常用选项

-f会将小写字母都转换为大写字母来进行比较，亦即忽略大小写

-c会检查文件是否已排好序，如果乱序，则输出第一个乱序的行的相关信息，最后返回1

-C会检查文件是否已排好序，如果乱序，不输出内容，仅返回1

-M会以月份来排序，比如JAN小于FEB等等

-b会忽略每一行前面的所有空白部分，从第一个可见字符开始比较。

### SED AWK案例展示

[root@localhost learn-script]# rpm -qai | head > ak

[root@localhost learn-script]# cat ak

Name : openssh-server

Version : 6.6.1p1

Release : 31.el7

Architecture: x86\_64

Install Date: Thu 05 Oct 2017 02:01:54 PM PDT

Group : System Environment/Daemons

Size : 943606

License : BSD

Signature : RSA/SHA256, Sun 20 Nov 2016 11:48:43 AM PST, Key ID 24c6a8a7f4a80eb5

Source RPM : openssh-6.6.1p1-31.el7.src.rpm

要查找size大小：

----------------------------

有如下方法：

awk -F: '/Size/{print $2}' ak

grep "Size" ak | awk -F: '{print $2}'

[root@localhost learn-script]# cat awktest

姓名,学号,地址,爱好

zhangpp,20150901,福建省福州市仓山区,跑步

zhangp,20150902,福建省泉州市南安市,登山

zhang123,20150903,福建省厦门市湖里区,羽毛球

lilin,20150902,福建省泉州市南安市,登山

chenyi,20150903,福建省厦门市湖里区,骑行

jiajia,20150902,福建省泉州市南安市,篮球

nihao,20150903,福建省厦门市湖里区,羽毛球

###将输出分隔符由逗号变为空格

[root@localhost learn-script]# awk -F, 'BEGIN{OFS=" "}{print $1,$2,$3,$4}' awktest

姓名 学号 地址 爱好

zhangpp 20150901 福建省福州市仓山区 跑步

zhangp 20150902 福建省泉州市南安市 登山

zhang123 20150903 福建省厦门市湖里区 羽毛球

lilin 20150902 福建省泉州市南安市 登山

chenyi 20150903 福建省厦门市湖里区 骑行

jiajia 20150902 福建省泉州市南安市 篮球

nihao 20150903 福建省厦门市湖里区 羽毛球

##此时并没有改变awktest内容，需要将输出屏幕内容打到临时文件，再将临时文件替换awktest，具体如下：

[root@localhost learn-script]# awk -F, 'BEGIN{OFS=" "}{print $1,$2,$3,$4}' awktest > tmp

[root@localhost learn-script]# mv tmp awktest

mv: overwrite ‘awktest’? y

[root@localhost learn-script]# cat awktest

姓名 学号 地址 爱好

zhangpp 20150901 福建省福州市仓山区 跑步

zhangp 20150902 福建省泉州市南安市 登山

zhang123 20150903 福建省厦门市湖里区 羽毛球

lilin 20150902 福建省泉州市南安市 登山

chenyi 20150903 福建省厦门市湖里区 骑行

jiajia 20150902 福建省泉州市南安市 篮球

nihao 20150903 福建省厦门市湖里区 羽毛球

#将学号姓名对调显示，此时不需要指定输入分隔符，因为awk默认分隔符就是空格

[root@localhost learn-script]# awk '{print $2,$1,$3,$4}' awktest

学号 姓名 地址 爱好

20150901 zhangpp 福建省福州市仓山区 跑步

20150902 zhangp 福建省泉州市南安市 登山

20150903 zhang123 福建省厦门市湖里区 羽毛球

20150902 lilin 福建省泉州市南安市 登山

20150903 chenyi 福建省厦门市湖里区 骑行

20150902 jiajia 福建省泉州市南安市 篮球

20150903 nihao 福建省厦门市湖里区 羽毛球

#将爱好信息提取出来，sort用于排序，将相同爱好的放在一起，用uniq删除相同的爱好

[root@localhost learn-script]# awk '{print $4}' awktest | sort | uniq

爱好

登山

篮球

羽毛球

跑步

骑行

#统计各个爱好的人数

[root@localhost learn-script]# awk '{print $4}' awktest | sort | uniq -c

1 爱好

2 登山

1 篮球

2 羽毛球

1 跑步

1 骑行

#统计各个爱好的人数，并将其输出重定向到tmp文件中

[root@localhost learn-script]# awk '{print $4}' awktest | sort | uniq -c > tmp

#输出爱好登山的人数

[root@localhost learn-script]# awk '/登山/{print $1}' tmp

2

#或者

[root@localhost learn-script]# awk '{print $4}' awktest | sort | uniq -c | awk '/登山/{print $1}'

2

---------------------------------------------------------

[root@localhost learn-script]# cat sedtest

姓名,学号,地址,爱好

zhangpp,20150901,福建省福州市仓山区,跑步

lilin,20150902,福建省泉州市南安市,登山

chenyi,20150903,福建省厦门市湖里区,骑行

jiajia,20150902,福建省泉州市南安市,篮球

nihao,20150903,福建省厦门市湖里区,羽毛球羽毛球

chener,20150903,福建省厦门市湖里区,骑行

jiaren,20150902,福建省泉州市南安市,足球

nini,20150903,福建省厦门市湖里区,羽毛球

#删除第2，3行

[root@localhost learn-script]# sed '2,3d' sedtest

姓名,学号,地址,爱好

chenyi,20150903,福建省厦门市湖里区,骑行

jiajia,20150902,福建省泉州市南安市,篮球

nihao,20150903,福建省厦门市湖里区,羽毛球羽毛球

chener,20150903,福建省厦门市湖里区,骑行

jiaren,20150902,福建省泉州市南安市,足球

nini,20150903,福建省厦门市湖里区,羽毛球

#删除第1行

[root@localhost learn-script]# sed '1d' sedtest

zhangpp,20150901,福建省福州市仓山区,跑步

lilin,20150902,福建省泉州市南安市,登山

chenyi,20150903,福建省厦门市湖里区,骑行

jiajia,20150902,福建省泉州市南安市,篮球

nihao,20150903,福建省厦门市湖里区,羽毛球羽毛球

chener,20150903,福建省厦门市湖里区,骑行

jiaren,20150902,福建省泉州市南安市,足球

nini,20150903,福建省厦门市湖里区,羽毛球

#删除2，3两行，并更改到源文件，有两个办法

[root@localhost learn-script]# sed '2,3d' sedtest > tmp

[root@localhost learn-script]# mv tmp sedtest

mv: overwrite ‘sedtest’? y

#或者

sed -i '2,3d' sedtest

#将sedtest中以chen打头的信息显示出来

[root@localhost learn-script]# sed -n '/^chen/p' sedtest

chenyi,20150903,福建省厦门市湖里区,骑行

chener,20150903,福建省厦门市湖里区,骑行

#如果不加-n，则显示如下

[root@localhost learn-script]# sed '/^chen/p' sedtest

姓名,学号,地址,爱好

chenyi,20150903,福建省厦门市湖里区,骑行

chenyi,20150903,福建省厦门市湖里区,骑行

jiajia,20150902,福建省泉州市南安市,篮球

nihao,20150903,福建省厦门市湖里区,羽毛球羽毛球

chener,20150903,福建省厦门市湖里区,骑行

chener,20150903,福建省厦门市湖里区,骑行

jiaren,20150902,福建省泉州市南安市,足球

nini,20150903,福建省厦门市湖里区,羽毛球

#提取出爱好信息，这个案例中通过管道将sed和awk结合在一起

[root@localhost learn-script]# sed -n '/^chen/p' sedtest | awk -F, '{print $4}'

骑行

骑行

#将nihao 的爱好羽毛球改为睡觉

[root@localhost learn-script]# sed -n '/nihao/s/羽毛球/睡觉/p' sedtest

nihao,20150903,福建省厦门市湖里区,睡觉羽毛球

[root@localhost learn-script]# sed -n '/nihao/s/羽毛球/睡觉/pg' sedtest

nihao,20150903,福建省厦门市湖里区,睡觉睡觉

----------------------------------------------------------------------------

[root@localhost learn-script]# date

Tue Mar 27 14:51:22 PDT 2018

[root@localhost learn-script]# date | awk '{print $6$2$1}'

2018MarTue

[root@localhost learn-script]# date | awk '{print $6,$2,$1}'

2018 Mar Tue

[root@localhost learn-script]#