

统计分析与数据挖掘

Data->information->knowledge->intelligen

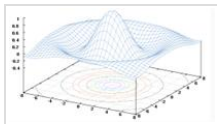


目录视图

摘要视图

RSS 订阅

个人资料



statdm

[+ 加关注](#) [发私信](#)

访问： 135374次

积分： 2123

等级： **BLOG > 5**

排名： 第7175名

原创： 75篇 转载： 2篇

译文： 0篇 评论： 24条

博主介绍

目前任职于上海某大型上市软件公司，主要负责统计分析&数据挖掘&数学优化三方面的产品开发及与此相关的项目支持。
研究领域：并行计算、大数据运算、统计&挖掘

文章分类

工具使用类 (27)
统计分析与数据挖掘 (25)
项目管理 (9)
云计算与云存储 (1)
linux 相关 (16)
游戏人生 (1)
R (2)
SAS (3)
matlab (1)
c/c++ (4)

文章存档

2013年01月 (4)
2012年12月 (4)
2012年11月 (6)
2012年08月 (2)
2012年07月 (25)

[展开](#)

阅读排行

Word 2007 如何设置多级 (8548)
丽江游玩攻略1 (7417)
典型相关分析相关资料 (6947)

[2015年4月微软MVP申请](#) [《京东技术解密》有奖试读，礼品大放送](#) [“我的2014”年度征文活动火爆开启](#) [CSDN 2014博客之星](#)

原 linux查找技巧： find grep xargs

分类： linux 相关

2012-07-05 14:54 6074人阅读 评论(0) 收藏 举报

linux

正则表达式

mina

shell

less

class

技术问答

在当前目录下所有.cpp文件中查找efg函数

find . -name "*.cpp" | [快速回复](#) efg'

xargs展开find获得的结果，使其作为grep的参数

另外 rm mv等命令对大量文件操作时报错 -bash: /bin/rm: Argument list too long也可用xargs 解决
删除当前目录下所有.cpp文件

find . -name "*.cpp" | xargs rm

////////////////////////////////////

find [起始目录] 寻找条件 操作

还有种表述方式：find PATH OPTION [-exec COMMAND { } \;]

因为find命令会根据我们给的option,也就是寻找条件从我们给出的目录开始对其中文件及其下子目录中的文件进行递归搜索，所以我觉的这个地方说是“起始目录”是非常好的。

该命令中的寻找条件可以是一个用逻辑运算符 not、and、or 组成的复合条件。逻辑运算符 and、or、not 的含义为：

(1) and：逻辑与，在命令中用“-a”表示，是系统缺省的选项，表示只有当所给的条件都满足时，寻找条件才算满足。例如：

find -name 'tmp' -xtype c -user 'inin'

% 该命令寻找三个给定条件都满足的所有文件

(2) or：逻辑或，在命令中用“-o”表示。该运算符表示只要所给的条件中有一个满足时，寻找条件就算满足。例如：

find -name 'tmp' -o -name 'mina'

% 该命令查询文件名为'tmp'或是匹配'mina*'的所有文件。

(3) not：逻辑非，在命令中用“!”表示。该运算符表示查找不满足所给条件的文件。例如：

find ! -name 'tmp'

% 该命令查询文件名不是'tmp'的所有文件。

需要说明的是：当使用很多的逻辑选项时，可以用括号把这些选项括起来。为了避免Shell本身对括号引起误解，在括号前需要加转义字符“\”来去除括号的意义。例：

find \(-name 'tmp' -xtype c -user 'inin' \)

我觉的现在我应该说下出了查询条件，在find中的option的内容了：

在option中，具体有参数：

-name '字串' 查找文件名匹配所给字串的所有文件，字串内可用通配符 *、?、[]。

-lname '字串' 查找文件名匹配所给字串的所有符号链接文件，字串内可用通配符 *、?、[]。

-gid n 查找属于ID号为 n 的用户组的所有文件。

-uid n 查找属于ID号为 n 的用户的所有文件。

-group '字串' 查找属于用户组名为所给字串的所有的文件。

-user '字串' 查找属于用户名为所给字串的所有的文件。

-empty 查找大小为 0 的目录或文件。

linux查找技巧: find grep (6074)

Fedora 添加国内源 (soh) (5978)

徒弟离职引发的思考 (4434)

打开outlook ,提示outlook (4310)

configure: error: C++ pre (4295)

逻辑回归与决策树在分类 (3850)

在R 语言中调用关联规则 (3783)

评论排行

fedora16 下安装 gcc4.7. (2)

决策树 & SAS (2)

如何做一个优秀的eader (2)

vmware与virtualbox的网: (2)

另眼看待变量间多重共线 (2)

徒弟离职引发的思考 (2)

老板为什么会被下属“反噬 (2)

hadoop/mapreduce-矩阵 (1)

nvidia driver for linux : ht (1)

SAS逻辑回归实例 (1)

推荐文章

* Qt Quick里的粒子系统

* Android proguard 详解

* 我的2014---感悟程序员职场

* 前端开发规范之项目架构

* 我的2014年总结 - 一些失败的面试经历

* 最简单的基于FFmpeg的编码器-纯净版

最新评论

SAS逻辑回归实例
budingningmeng: 要怎样才能看
图啊

徒弟离职引发的思考
vitha: 曾经带过最大的团队81人
(分成好几个小组), 遇到过各
种各样的问题, 现在回想起来,
当时有些解决方法很幼...

徒弟离职引发的思考
vitha: 楼主不必自责, 任何行业人
才流动都是正常的, 特别是现在
这个社会环境。作为一个部门负
责人来看, 我们需要的...

需求分析中的注意点
vitha: 这个很中肯, 楼主参加的项
目也是敏捷模式的吗? 共同学习
吧, 目前也是参与这类项目的管
理工作,

另眼看待变量间多重共线性
huangxia73: 思路很好!

丽江游玩攻略1
lianyu53653: 想要去云南旅游
吗? 丽江、大理、香格里拉、西
双版纳或者是腾冲? 登录旅程网
www.4006888808....

hadoop/mapreduce-矩阵乘法的§
YJun2012: 楼主, 你这个是Linux
环境下还Windows下, 还有路径
能输入像D: /test.txt这种吗?

如何做一个优秀的eader
chao1118: 博主的文章很好, 醍
醐灌顶, 感谢分享!

统计分析与数据挖掘&数学优化&c
xiaohang20103: 表示人大经济论
坛是不错的, 经常提供各种软件
和资料, 在数据分析上面不错

决策树 & SAS

-path '字串' 查找路径名匹配所给字串的所有文件, 字串内可用通配符*、?、[]。

-perm 权限 查找具有指定权限的文件和目录, 权限的表示可以如711, 644。

-size n[bckw] 查找指定文件大小的文件, n 后面的字符表示单位, 缺省为 b, 代表512字节的块。

-type x 查找类型为 x 的文件, x 为下列字符之一:

b 块设备文件

c 字符设备文件

d 目录文件

p 命名管道(FIFO)

f 普通文件

l 符号链接文件(symbolic links)

s socket文件

-xtype x 与 -type 基本相同, 但只查找符号链接文件。

以时间为条件查找

-amin n 查找n分钟以前被访问过的所有文件。

-atime n 查找n天以前被访问过的所有文件。

-cmin n 查找n分钟以前文件状态被修改过的所有文件。

-ctime n 查找n天以前文件状态被修改过的所有文件。

-mmin n 查找n分钟以前文件内容被修改过的所有文件。

-mtime n 查找n天以前文件内容被修改过的所有文件。

-print: 将搜索结果输出到标准输出。

例子: 在root以及子目录查找不包括目录/root/bin的, greek用户的, 文件类型为普通文件的, 3天之前的名为test-find.c的文件, 并将结构输出, find命令如下:

find / -name "test-find.c" -type f -mtime +3 -user greek -prune /root/bin -print

当然在这其中, -print是一个默认选项, 我们不必刻意去配置它。

我们再看一下exec选项:

-exec: 对搜索的结构指令指定的shell命令。注意格式要正确: "-exec 命令 {} \;"

在}和\之间一定要有空格才行;

{ }表示命令的参数即为所找到的文件;命令的末尾必须以“\;"结束。

例子: 对上述例子搜索出来的文件进行删除操作, 命令如下:

find / -name "test-find.c" -type f -mtime +3 -user greek -prune /root/bin -exec rm {} \;

find命令指令实例:

find . - name 'main*' - exec more {} \;

% 查找当前目录中所有以main开头的文件, 并显示这些文件的内容。

find . \(- name a.out - o - name *.o \) > - atime +7 - exec rm {} \;

% 删除当前目录下所有一周之内没有被访问过的a .out或*.o文件。

% 命令中的"."表示当前目录, 此时 find 将从当前目录开始, 逐个在其子目录中查找满足后面指定条件的文件。

% "(" 和 ")" 表示括号(), 其中的 "\" 称为转义符。之所以这样写是由于对 Shell 而言, (和)另有不同的含义, 而不是这里的用于组合条件的用途。

% "-name a.out" 是指要查找名为a.out的文件;

% "-name *.o" 是指要查找所有名字以 .o 结尾的文件。

这两个 -name 之间的 -o 表示逻辑或(or), 即查找名字为a.out或名字以 .o结尾的文件。

% find命令在当前目录及其子目录下找到这样的文件之后, 再进行判断, 看其最后访问时间 是否在7天以前(条件 -atime +7), 若是, 则对该文件执行命令 rm(-exec rm {} \;)。

其中 { } 代表当前查到的符合条件的文件名, \;则是语法所要求的。

% 上述命令中第一行的最后一个\是续行符。当命令太长而在一行写不下时, 可输入一个\, 之后系统将显示一个 >, 指示用户继续输入命令。

////////////////////////////////////

linux grep命令

<div fc05="" fc11="" nbw-blog="" ztag="" js-fs2="">

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]

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:

- \: 忽略正则表达式中特殊字符的原有含义。
- ^: 匹配正则表达式的开始行。
- \$: 匹配正则表达式的结束行。
- \<: 从匹配正则表达式的行开始。
- \>: 到匹配正则表达式的行结束。
- []: 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。
- .: 所有的单个字符。
- *: 有字符, 长度可以为0。

4.grep命令使用简单实例

```
$ grep 'test' d*
```

显示所有以d开头的文件中包含 test的行。

```
$ grep 'test' aa bb cc
```

显示在aa, bb, cc文件中匹配test的行。

```
$ grep '[a-z]\{5\}' aa
```

显示所有包含每个字符串至少有5个连续小写字母的字符串的行。

```
$ grep 'w(es)t.*1' aa
```

如果west被匹配, 则es就被存储到内存中, 并标记为1, 然后搜索任意个字符(.*) , 这些字符后面紧跟着 另外一个es(1), 找到就显示该行。如果用egrep或grep -E, 就不用“\”号进行转义, 直接写成w(es)t.*1就可以了。

5.grep命令使用复杂实例

假设您正在/usr/src/Linux/Doc目录下搜索带字符串‘magic’的文件:

```
$ grep magic /usr/src/Linux/Doc/*
```

```
sysrq.txt:* How do I enable the magic SysRQ key?
```

```
sysrq.txt:* How do I use the magic SysRQ key?
```

其中文件‘sysrq.txt’包含该字符串, 讨论的是 SysRQ 的功能。

默认情况下, ‘grep’只搜索当前目录。如果 此目录下有许多子目录, ‘grep’会以如下形式列出:

```
grep: sound: Is a directory
```

这可能会使‘grep’的输出难于阅读。这里有两种解决的办法:

明确要求搜索子目录: `grep -r`

或忽略子目录: `grep -d skip`

如果有很多 输出时, 您可以通过管道将其转到‘less’上阅读:

```
$ grep magic /usr/src/Linux/Documentation/* | less
```

这样, 您就可以更方便地阅读。

有一点要注意, 您必需提供一个文件过滤方式(搜索全部文件的话用 *)。如果您忘了, ‘grep’会一直等着, 直到该程序被中断。如果您遇到了这样的情况, 按 <CTRL c> , 然后再试。

下面还有一些有意思的命令行参数:

grep -i pattern files : 不区分大小写地搜索。默认情况区分大小写,

grep -l pattern files : 只列出匹配的文件名,

grep -L pattern files : 列出不匹配的文件名,

grep -w pattern files : 只匹配整个单词, 而不是字符串的一部分(如匹配‘magic’, 而不是‘magical’),

grep -C number pattern files : 匹配的上下文分别显示[number]行,

grep pattern1 | pattern2 files : 显示匹配 pattern1 或 pattern2 的行,

grep pattern1 files | grep pattern2 : 显示既匹配 pattern1 又匹配 pattern2 的行。

grep -n pattern files 即可显示行号信息

grep -c pattern files 即可查找总行数

这里还有些用于搜索的特殊符号:

\< 和 \> 分别标注单词的开始与结尾。

例如:

grep man * 会匹配 ‘Batman’、‘manic’、‘man’等,

grep '\<man' * 匹配 ‘manic’和‘man’, 但不是 ‘Batman’,

grep '\<man\>' 只匹配 ‘man’, 而不是 ‘Batman’或‘manic’等其他的字符串。

‘^’: 指匹配的字符串在行首,

‘\$’: 指匹配的字符串在行尾,

Grep 命令 用法大全

1、 参数:

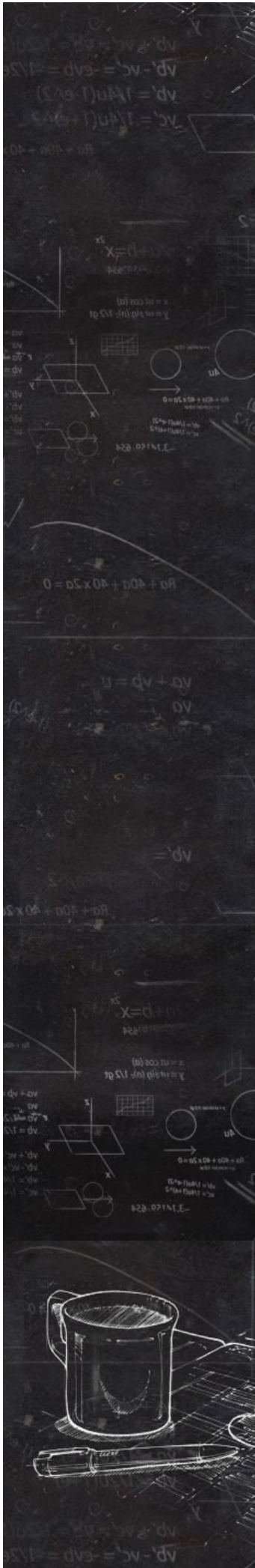
-I : 忽略大小写

-c : 打印匹配的行数

-l : 从多个文件中查找包含匹配项

-v : 查找不包含匹配项的行

-n : 打印包含匹配项的行和行标



2、RE（正则表达式）

\ 忽略正则表达式中特殊字符的原有含义
^ 匹配正则表达式的开始行
\$ 匹配正则表达式的结束行
\< 从匹配正则表达式的行开始
\> 到匹配正则表达式的行结束
[] 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
. 所有的单个字符
* 所有字符，长度可以为0

3、举例

```
# ps -ef | grep in.telnetd  
root 19955 181 0 13:43:53 ? 0:00 in.telnetd
```

more size.txt size文件的内容

b124230
b034325
a081016
m7187998
m7282064
a022021
a061048
m9324822
b103303
a013386
b044525
m8987131
B081016
M45678
B103303
BADc2345

more size.txt | grep '[a-b]' 范围；如[A-Z]即A，B，C一直到Z都符合要求

b124230
b034325
a081016
a022021
a061048
b103303
a013386
b044525

more size.txt | grep '[a-b]'

b124230
b034325
a081016
m7187998
m7282064
a022021
a061048
m9324822
b103303
a013386
b044525
m8987131
B081016
M45678
B103303
BADc2345

more size.txt | grep 'b' 单个字符；如[A] 即A符合要求

b124230
b034325
b103303
b044525

more size.txt | grep '[bB]'

b124230
b034325
b103303
b044525
B081016
B103303
BADc2345

grep 'root' /etc/group

```
root::0:root
bin::2:root,bin,daemon
sys::3:root,bin,sys,adm
adm::4:root,adm,daemon
uucp::5:root,uucp
mail::6:root
tty::7:root,tty,adm
lp::8:root,lp,adm
nuucp::9:root,nuucp
daemon::12:root,daemon
```

```
# grep '^root' /etc/group 匹配正则表达式的开始行
root::0:root
```

```
# grep 'uucp' /etc/group
uucp::5:root,uucp
nuucp::9:root,nuucp
```

```
# grep '\<uucp' /etc/group
uucp::5:root,uucp
```

```
# grep 'root$' /etc/group 匹配正则表达式的结束行
root::0:root
mail::6:root
```

```
# more size.txt | grep -i 'b1..*3' -i : 忽略大小写
```

```
b124230
b103303
B103303
```

```
# more size.txt | grep -iv 'b1..*3' -v : 查找不包含匹配项的行
```

```
b034325
a081016
m7187998
m7282064
a022021
a061048
m9324822
a013386
b044525
m8987131
B081016
M45678
BADc2345
```

```
# more size.txt | grep -in 'b1..*3'
1:b124230
9:b103303
15:B103303
```

```
# grep '$' /etc/init.d/nfs.server | wc -l
128
```

```
# grep '\$' /etc/init.d/nfs.server | wc -l 忽略正则表达式中特殊字符的原有含义
```

```
15
# grep '\$' /etc/init.d/nfs.server
case "$1" in
>/tmp/sharetab.$$
[ "x$fstype" != xnfs ] &&
echo "$path\t$res\t$fstype\t$opts\t$desc"
>>/tmp/sharetab.$$
/usr/bin/touch -r /etc/dfs/sharetab /tmp/sharetab.$$
/usr/bin/mv -f /tmp/sharetab.$$ /etc/dfs/sharetab
if [ -f /etc/dfs/dfstab ] && /usr/bin/egrep -v '^[ ]*(#|$)'
if [ $startnfsd -eq 0 -a -f /etc/rmmount.conf ] &&
if [ $startnfsd -ne 0 ]; then
elif [ ! -n "$_INIT_RUN_LEVEL" ]; then
while [ $wtime -gt 0 ]; do
wtime=`expr $wtime - 1`
if [ $wtime -eq 0 ]; then
echo "Usage: $0 { start | stop }"
```

```
# more size.txt
```

the test file
their are files
The end

```
# grep 'the' size.txt
the test file
their are files
```

```
# grep '\<the' size.txt
the test file
their are files
```

```
# grep 'the\>' size.txt
the test file
```

```
# grep '\<the\>' size.txt
the test file
```

```
# grep '\<[Tt]he\>' size.txt
the test file
```

1,简介

使用正则表达式的一个多用途文本搜索工具.这个`php?name=%C3%FC%C1%EE" onclick="tagshow(event)" class="t_tag">命令`本来是ed行编辑器中的一个php?name=%C3%FC%C1%EE" onclick="tagshow(event)" class="t_tag">命令/过滤器:
g/re/p -- global - regular expression - print.

基本格式

grep pattern [file...]

(1)grep 搜索字符串 [filename]

(2)grep 正则表达式 [filename]

在文件中搜索所有 pattern 出现的位置, pattern 既可以是要搜索的字符串,也可以是一个正则表达式.
注意: 在输入要搜索的字符串时最好使用双引号/而在模式匹配使用正则表达式时, 注意使用单引号

2,grep的选项

- c 只输出匹配行的计数
- i 不区分大小写 (用于单字符)
- n 显示匹配的行号
- v 不显示不包含匹配文本的所以有行
- s 不显示错误信息
- E 使用扩展正则表达式

更多的选项请查看: man grep

3,常用grep实例

(1)多个文件查询

```
grep "sort" *.doc #见文件名的匹配
```

(2)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
```

(4)显示非匹配的行

```
grep -vn "48" data.doc #输出所有不包含48的行
```

(5)大小写敏感

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

4, 正则表达式的应用

(1)正则表达式的应用 (注意: 最好把正则表达式用单引号括起来)

```
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 -E '219|216' data.doc
```

(4) ...

这需要在实践中不断应用和总结，熟练掌握正则表达式。

5, 使用类名

可以使用国际模式匹配的类名：

`[[[:upper:]]]` `[A-Z]`

`[[[:lower:]]]` `[a-z]`

`[[[:digit:]]]` `[0-9]`

`[[[:alnum:]]]` `[0-9a-zA-Z]`

`[[[:space:]]]` 空格或tab

`[[[:alpha:]]]` `[a-zA-Z]`

(1)使用

`grep '5[[[:upper:]]][[:upper:]]' data.doc` #查询以5开头以两个大写字母结尾的行



上一篇 科普-文本挖掘（文本分类）流程

下一篇 使用yum下载而不安装rpm包

顶

0

踩

0

主题推荐

linux

编辑器

实例

阅读

应用

猜你在找

Sabre

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 070598 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved