

flyor

昵称: flyor  
园龄: 3年10个月  
粉丝: 16  
关注: 3  
+加关注

< 2020年11月 >						
日	一	二	三	四	五	六
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5
6	7	8	9	10	11	12

搜索

常用链接

我的随笔  
我的评论  
我的参与  
最新评论  
我的标签  
更多链接

我的标签

javascript(6)  
js(6)  
Linux(3)  
入门(2)  
上取整(1)  
网络基础(1)  
文本处理(1)  
下取整(1)  
协议(1)  
BOM(1)  
更多

随笔分类

http协议(1)  
JavaScript(6)  
Linux(4)  
操作系统内核那些事(1)  
汇编语言(1)

博客园 首页 新随笔 联系 管理 订阅 XML

随笔- 15 文章- 2 评论- 6

linux中grep命令的用法

作为linux中最为常用的三大文本 ( awk · sed · grep ) 处理工具之一，掌握好其用法是很有必要的。

首先谈一下grep命令的常用格式为：grep [选项] “模式” [文件]

grep家族总共有三个：grep · egrep · fgrep。

常用选项：

- E ：开启扩展 ( Extend ) 的正则表达式。
- i ：忽略大小写 ( ignore case ) 。
- v ：反过来 ( invert ) ，只打印没有匹配的，而匹配的反而不打印。
- n ：显示行号
- w ：被匹配的文本只能是单词，而不能是单词中的某一部分，如文本中有liker，而我搜寻的只是like，就可以使用-w选项来避免匹配liker
- c ：显示总共有多少行被匹配到了，而不是显示被匹配到的内容，注意如果同时使用-cv选项是显示有多少行没有被匹配到。
- o ：只显示被模式匹配到的字符串。
- color ：将匹配到的内容以颜色高亮显示。
- A n：显示匹配到的字符串所在的行及其后n行，after
- B n：显示匹配到的字符串所在的行及其前n行，before
- C n：显示匹配到的字符串所在的行及其前后各n行，context

计算机基础知识(1)  
数学(1)

## 随笔档案

2019年5月(1)  
2018年4月(1)  
2018年3月(5)  
2018年2月(1)  
2018年1月(1)  
2017年12月(1)  
2017年7月(3)  
2017年2月(2)

## 最新评论

1. Re:linux中grep命令的用法  
牛批  
--venchia
2. Re:linux中grep命令的用法  
谢谢  
--蓝色的螺丝钉
3. Re:linux中grep命令的用法  
牛批  
--饮水机管理员y
4. Re:JavaScript基础入门教程（一）  
从c#看过来的，很受用！很适合！  
--自古深情留不住
5. Re:linux中grep命令的用法  
非常感谢~好人一生平安。  
--1024hack

## 阅读排行榜

1. linux中grep命令的用法(291524)
2. 汇编语言系列教程之基础入门（一）(19653)
3. linux用户管理(10041)
4. vim的tab键设定(9411)
5. JavaScript基础入门教程（一）(7546)

## 评论排行榜

1. linux中grep命令的用法(5)
2. JavaScript基础入门教程（一）(1)

## 推荐排行榜

1. linux中grep命令的用法(15)
2. JavaScript基础入门教程（一）(1)
3. linux用户管理(1)
4. 汇编语言系列教程之基础入门（一）(1)

```
[root@compute demo]# grep "root" /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin
[root@compute demo]# grep -i "Root" /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin
[root@compute demo]# grep -n "root" /etc/passwd
1:root:x:0:0:root:/root:/bin/bash
10:operator:x:11:0:operator:/root:/sbin/nologin
[root@compute demo]# grep -vc "root" /etc/passwd
63
[root@compute demo]# grep -o "root" /etc/passwd
root
root
root
root
```

```
[root@compute demo]# grep -A 2 "core id" /proc/cpuinfo
core id      : 0
cpu cores    : 2
apicid       : 0
--
core id      : 1
cpu cores    : 2
apicid       : 1
[root@compute demo]# grep -B 2 "core id" /proc/cpuinfo
physical id  : 0
siblings     : 2
core id      : 0
--
physical id  : 0
siblings     : 2
core id      : 1
[root@compute demo]# grep -C 2 "core id" /proc/cpuinfo
physical id  : 0
siblings     : 2
core id      : 0
cpu cores    : 2
apicid       : 0
--
physical id  : 0
siblings     : 2
core id      : 1
cpu cores    : 2
apicid       : 1
```

## 模式部分：

1、直接输入要匹配的字符串，这个可以用fgrep ( fast grep ) 代替来提高查找速度，比如我要匹配一下hello.c文件中printf的个数：fgrep -c "printf" hello.c

2、使用基本正则表达式，下面谈关于基本正则表达式的使用：

匹配字符：

. ：任意一个字符。

[abc] ：表示匹配一个字符，这个字符必须是abc中的一个。

[a-zA-Z] ：表示匹配一个字符，这个字符必须是a-z或A-Z这52个字母中的一个。

[^123] : 匹配一个字符，这个字符是除了1、2、3以外的所有字符。

对于一些常用的字符集，系统做了定义：

[A-Za-z] 等价于 [[:alpha:]]

[0-9] 等价于 [[:digit:]]

[A-Za-z0-9] 等价于 [[:alnum:]]

tab,space 等空白字符 [[:space:]]

[A-Z] 等价于 [[:upper:]]

[a-z] 等价于 [[:lower:]]

标点符号 [[:punct:]]

```
[root@compute demo]# cat demo.c
#include <stdio.h>
int main()
{
    int hello=3;
    int world=4;
    int helloWorld=hello+world;
    printf("%d\n",helloWorld);
    return 0;
}
[root@compute demo]# grep "hello." demo.c
    int hello=3;
    int helloWorld=hello+world;
    printf("%d\n",helloWorld);
[root@compute demo]# grep "hello[[:upper:]]" demo.c
    int helloWorld=hello+world;
    printf("%d\n",helloWorld);
[root@compute demo]# grep "hello[^[:upper:]][[:digit:]]" demo.c
    int hello=3;
[root@compute demo]# grep "hell[a-z]" demo.c
    int hello=3;
    int helloWorld=hello+world;
    printf("%d\n",helloWorld);
[root@compute demo]# grep "hell[a-z][[:punct:]]" demo.c
    int hello=3;
    int helloWorld=hello+world;
```

匹配次数：

\{m,n\} : 匹配其前面出现的字符至少m次，至多n次。

\? : 匹配其前面出现的内容0次或1次，等价于\{0,1\}。

\* : 匹配其前面出现的内容任意次，等价于\{0,\}，所以 ".\*" 表述任意字符任意次，即无论什么内容全部匹配。

```
[root@compute demo]# grep "/.*sh" /etc/passwd
root:x:0:0:root:/root:/bin/bash
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
dirsrv:x:389:389:389-ds-base:/usr/share/dirsrv:/sbin/nologin
amandabackup:x:33:6:Amanda user:/var/lib/amanda:/bin/bash
setroubleshoot:x:385:385::/var/lib/setroubleshoot:/sbin/nologin
tomcat:x:91:91:Apache Tomcat:/usr/share/tomcat:/sbin/nologin
pkuser:x:17:17:Certificate System:/usr/share/pki:/sbin/nologin
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
compute:x:1000:1000:compute:/home/compute:/bin/bash
user3:x:505:0::/home/user3:/bin/tcsh
user5:x:5000:5000::/home/user5:/bin/bash
[root@compute demo]# grep "/.\{0,2\}sh" /etc/passwd
root:x:0:0:root:/root:/bin/bash
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
dirsrv:x:389:389:389-ds-base:/usr/share/dirsrv:/sbin/nologin
amandabackup:x:33:6:Amanda user:/var/lib/amanda:/bin/bash
tomcat:x:91:91:Apache Tomcat:/usr/share/tomcat:/sbin/nologin
pkuser:x:17:17:Certificate System:/usr/share/pki:/sbin/nologin
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
compute:x:1000:1000:compute:/home/compute:/bin/bash
user3:x:505:0::/home/user3:/bin/tcsh
user5:x:5000:5000::/home/user5:/bin/bash
[root@compute demo]# grep -w "/.\{0,2\}sh" /etc/passwd
root:x:0:0:root:/root:/bin/bash
amandabackup:x:33:6:Amanda user:/var/lib/amanda:/bin/bash
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
compute:x:1000:1000:compute:/home/compute:/bin/bash
user3:x:505:0::/home/user3:/bin/tcsh
user5:x:5000:5000::/home/user5:/bin/bash
```

#### 位置锚定：

^ ：锚定行首

\$ ：锚定行尾。技巧：“^\$”用于匹配空白行。

\b或\<：锚定单词的词首。如“\blike”不会匹配alike，但是会匹配liker

\b或\>：锚定单词的词尾。如“\blike\b”不会匹配alike和liker，只会匹配like

\B ：与\b作用相反。

```
[root@compute ~]# grep "h" /etc/passwd
root:x:0:0:root:/root:/bin/bash
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
ods:x:999:998:softsm private keys owner:/var/lib/softsm:/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
hacluster:x:189:189:cluster user:/home/hacluster:/sbin/nologin
dirsrv:x:389:389:389-ds-base:/usr/share/dirsrv:/sbin/nologin
amandabackup:x:33:6:Amanda user:/var/lib/amanda:/bin/bash
saslauth:x:387:76:Saslauthd user:/run/saslauthd:/sbin/nologin
setroubleshoot:x:385:385:/var/lib/setroubleshoot:/sbin/nologin
memcached:x:384:384:Memcached daemon:/run/memcached:/sbin/nologin
chrony:x:383:383:/var/lib/chrony:/sbin/nologin
hsqldb:x:96:96:/var/lib/hsqldb:/sbin/nologin
tomcat:x:91:91:Apache Tomcat:/usr/share/tomcat:/sbin/nologin
pkuser:x:17:17:Certificate System:/usr/share/pki:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
dovecot:x:380:378:Dovecot's unauthorized user:/usr/libexec/dovecot:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
compute:x:1000:1000:compute:/home/compute:/bin/bash
user4:x:1013:1013:/home/user4:/sbin/nologin
user3:x:505:0:/home/user3:/bin/tcsh
user5:x:5000:5000:/home/user5:/bin/bash
[root@compute ~]# grep "h$" /etc/passwd
root:x:0:0:root:/root:/bin/bash
amandabackup:x:33:6:Amanda user:/var/lib/amanda:/bin/bash
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
compute:x:1000:1000:compute:/home/compute:/bin/bash
user3:x:505:0:/home/user3:/bin/tcsh
user5:x:5000:5000:/home/user5:/bin/bash
```

```
[root@compute ~]# grep "sh" /etc/passwd
root:x:0:0:root:/root:/bin/bash
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
dirsrv:x:389:389:389-ds-base:/usr/share/dirsrv:/sbin/nologin
amandabackup:x:33:6:Amanda user:/var/lib/amanda:/bin/bash
setroubleshoot:x:385:385:/var/lib/setroubleshoot:/sbin/nologin
tomcat:x:91:91:Apache Tomcat:/usr/share/tomcat:/sbin/nologin
pkuser:x:17:17:Certificate System:/usr/share/pki:/sbin/nologin
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
compute:x:1000:1000:compute:/home/compute:/bin/bash
user3:x:505:0:/home/user3:/bin/tcsh
user5:x:5000:5000:/home/user5:/bin/bash
[root@compute ~]# grep "\<sh" /etc/passwd
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
dirsrv:x:389:389:389-ds-base:/usr/share/dirsrv:/sbin/nologin
tomcat:x:91:91:Apache Tomcat:/usr/share/tomcat:/sbin/nologin
pkuser:x:17:17:Certificate System:/usr/share/pki:/sbin/nologin
[root@compute ~]# grep "\Bsh\b" /etc/passwd
root:x:0:0:root:/root:/bin/bash
amandabackup:x:33:6:Amanda user:/var/lib/amanda:/bin/bash
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
compute:x:1000:1000:compute:/home/compute:/bin/bash
user3:x:505:0:/home/user3:/bin/tcsh
user5:x:5000:5000:/home/user5:/bin/bash
```

#### 分组及引用：

\(string\) ：将string作为一个整体方便后面引用

\1 ：引用第1个左括号及其对应的右括号所匹配的内容。

\2 ：引用第2个左括号及其对应的右括号所匹配的内容。

\n ：引用第n个左括号及其对应的右括号所匹配的内容。

```
[root@compute ~]# grep "^([[:alpha:]]\)*.*$" /etc/passwd #以相同字母开始并结尾的行
nobody:x:99:99:Nobody:/sbin/nologin
named:x:25:25:Named:/var/named:/sbin/nologin
ntp:x:38:38:/etc/ntp:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
nova:x:162:162:OpenStack Nova Daemons:/var/lib/nova:/sbin/nologin
neutron:x:379:377:OpenStack Neutron Daemons:/var/lib/neutron:/sbin/nologin
```

3、扩展的 ( **Extend** ) 正则表达式 ( 注意要使用扩展的正则表达式要加 **-E** 选项，或者直接使用 **egrep** )：

**匹配字符**：这部分和基本正则表达式一样

**匹配次数**：

**\***：和基本正则表达式一样

**?**：基本正则表达式是 `\?`，二这里没有 `\`。

**{m,n}**：相比基本正则表达式也是没有了 `\`。

**+**：匹配其前面的字符至少一次，相当于 `{1,}`。

**位置锚定**：和基本正则表达式一样。

**分组及引用**：

**(string)**：相比基本正则表达式也是没有了 `\`。

**\1**：引用部分和基本正则表达式一样。

**\n**：引用部分和基本正则表达式一样。

**或者**：

**a|b**：匹配a或b，注意a是指 `|` 的左边的整体，b也同理。比如 `C|cat` 表示的是C或cat，而不是Cat或cat，如果要表示Cat或cat，则应该写为 `(C|c)at`。记住 **(string)** 除了用于引用还用于分组。

注1：默认情况下，正则表达式的匹配工作在贪婪模式下，也就是说它会尽可能长地去匹配，比如某一行有字符串 `abacb`，如果搜索内容为 `"a.*b"` 那么会直接匹配 `abacb`这个串，而不会只匹配 `ab`或`acb`。

注2：所有的正则字符，如 `[`、`*`、`\`、`(` 等，若要搜索 `*`，而不是想把 `*` 解释为重复先前字符任意次，可以使用 `\*` 来转义。

下面用一个练习来结束本次grep的学习：

在网络配置文件 `/etc/sysconfig/network-scripts/ifcfg-ens33` 中检索出所有的 IP

1、检索出 0-255的范围

```
[root@compute demo]# egrep "[0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5]" \
/etc/sysconfig/network-scripts/ifcfg-ens33
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="ens33"
UUID="2f65fbb3-deb8-46e8-b0a9-699085f45566"
DEVICE="ens33"
IPADDR="172.29.132.181"
PREFIX="24"
GATEWAY="172.29.132.1"
DNS1="202.102.154.3"
IPV6_PEERDNS="yes"
IPV6_PEERROUTES="yes"
IPV6_PRIVACY="no"
```

2、由0-255的数字组合成IP

```
[root@compute demo]# egrep "\b([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.\.\  
> ([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.\.\  
> ([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.\.\  
> ([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\b" \  
> /etc/sysconfig/network-scripts/ifcfg-ens33  
IPADDR="172.29.132.181"  
GATEWAY="172.29.132.1"  
DNS1="202.102.154.3"
```

3、简化

```
[root@compute demo]# egrep "\b(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}\.  
> ([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\b" \  
> /etc/sysconfig/network-scripts/ifcfg-ens33  
IPADDR="172.29.132.181"  
GATEWAY="172.29.132.1"  
DNS1="202.102.154.3"
```

分类：[Linux](#)

标签：[文本处理](#)，[Linux](#)，[入门](#)

好文要顶

关注我

收藏该文

[flyor](#)  
关注 - 3  
粉丝 - 16

[+加关注](#)

150

« 上一篇：[vim的tab键设定](#)  
» 下一篇：[汇编语言系列教程之基础入门（一）](#)

posted @ 2017-02-17 18:35 flyor 阅读(291544) 评论(5) 编辑 收藏

评论

#1楼 2019-04-07 13:18   杏帘在望	
相当不错的文章，对grep的基本用法解析的很清楚	支持(0) 反对(0)
#2楼 2019-04-16 21:35   1024hack	
非常感谢~好人一生平安。	支持(0) 反对(0)
#3楼 2019-12-05 10:41   饮水机管理员y	
牛批	支持(0) 反对(0)
#4楼 2020-01-10 22:53   蓝色的螺丝钉	
谢谢	支持(1) 反对(0)
#5楼 2020-06-01 18:14   vencia	
牛批	支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

登录后才能发表评论，立即 [登录](#) 或 [注册](#)， [访问 网站首页](#)

博客园派送云上免费午餐，AWS注册立享12个月免费套餐

- 【推荐】News: 大型组态、工控、仿真、CADGIS 50万行VC++源码免费下载
- 【推荐】#双11学习节# IT技能、厂商认证、软考、PMP，多款好课钜惠
- 【推荐】博客园 & 陌上花开HIMMR 给单身的程序员小哥哥助力脱单啦~