

向架构师的方向，缓慢前行

博客园

首页

新随笔

联系

订阅

管理

公告

昵称: [goodbay说拜拜](#)

园龄: 1年2个月

粉丝: 2

关注: 1

[-取消关注](#)

< 2023年6月 >						
日	一	二	三	四	五	六
28	29	30	31	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

常用链接

[我的随笔](#)

[我的评论](#)

[我的参与](#)

[最新评论](#)

[我的标签](#)

随笔分类 (176)

[实战案例\(24\)](#)

[Linux-系统应用-命令\(29\)](#)

[Linux练习\(9\)](#)

[shell脚本\(3\)](#)

[shell脚本编写\(7\)](#)

[DNS 服务模块\(3\)](#)

[防火墙模块\(3\)](#)

[VPN配置流程\(3\)](#)

[MySQL\(24\)](#)

[运维自动化之ANSIBLE\(4\)](#)

[LAMP\(3\)](#)

[日志服务管理\(2\)](#)

[网络文件共享\(6\)](#)

[高可用集群\(4\)](#)

[NGINX\(13\)](#)

[tomcat\(10\)](#)

随笔 - 152 文章 - 0 评论 - 0 阅读 - 23477

Linux文本处理三剑客

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

格式：

```
grep [OPTIONS] PATTERN [FILE...]
```

常见选项

	
-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
-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
-w	匹配整个单词
-E	使用ERE, 相当于egrep
-F	不支持正则表达式, 相当于
-f file	根据模式文件处理
-r	递归目录, 但不处理软链接
-R	递归目录, 但处理软链接

分区利用率最大的值

```
[root@centos8 ~]#df | grep '^/dev/sd' | tr -s :  
[root@centos8 ~]#df |grep '^/dev/sd' |grep  
[root@centos8 ~]#df |grep '^/dev/sd' |grep  
13
```

反向代理HAproxy(5)
redis(11)
docker(10)
kubernetes(1)
zabbix监控(2)

哪个IP和当前主机连接数最多的前三位

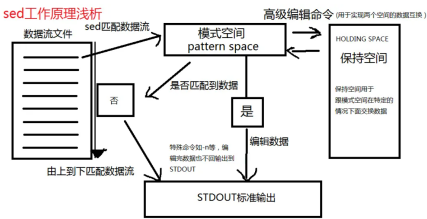
```
[root@centos8 ~]# ss -nt | grep "^ESTAB" | tr
3 10.0.0.1
1 172.16.4.100
1 172.16.31.188
```

连接状态的统计

```
[root@wang-liyun-pc ~]# ss -nta | grep -v '
7 ESTAB
4 LISTEN
7 TIME-WAIT
[root@wang-liyun-pc ~]# ss -nta | tail -n +:
3 ESTAB
4 LISTEN
12 TIME-WAIT
```

2、文本处理之sed

Sed是从文件或管道中读取一行，处理一行，输出一行；再读取一行，再处理一行，再输出一行，直到最后一行。每当处理一行时，把当前处理的行存储在临时缓冲区中，称为模式空间（Pattern Space），接着用sed命令处理缓冲区中的内容，处理完成后，把缓冲区的内容送往屏幕。



2.1.1 sed 基本用法

格式

```
sed [option]... 'script;script;...' [inputfile]
```

常用选项

-n

不输出模式空间内容到屏幕，即不自动

-e

多点编辑

-f FILE

从指定文件中读取编辑脚本

-r, -E

使用扩展正则表达式

-i.bak

备份文件并原处编辑

-s

将多个文件视为独立文件，而不是单个

#说明:

-ir

不支持

-i -r

支持

-ri

支持

-ni

会清空文件

script格式

'地址命令'

地址格式

1. 不给地址：对全文进行处理

2. 单地址：
#：指定的行，\$：最后一行
/pattern/：被此处模式所能够匹配到的每一行

3. 地址范围：
#, # #从#行到第#行，3, 6 从3到6
#, +# #从#行到+#行，3, +4 表示从3行到第4行的下一行
/pat1/, /pat2/ #正则表达式
#, /pat/
/pat/, #

4. 步进：~
1~2 奇数行
2~2 偶数行

命令:

p 打印当前模式空间内容，追加到默认输出:

Ip 忽略大小写输出

d 删除模式空间匹配的行，并立即启用下一行

a [\]text 在指定行后面追加文本，支持使用\n实现换行

i [\]text	在行前面插入文本
c [\]text	替换行为单行或多行文本
w file	保存模式匹配的行至指定文件
r file	读取指定文件的文本至模式空间中匹配到
=	为模式空间中的行打印行号
!	模式空间中匹配行取反处理
q	结束或退出sed

查找替代

s/pattern/string/修饰符	查找替换, 支持使用其它替换修饰符:
g	行内全局替换
p	显示替换成功的行
w /PATH/FILE	将替换成功的行保存至文件中
I, i	忽略大小写

示例:

```
[11:20:54 root@centos8 ~]#sed ' ' /etc/issue
\S
Kernel \r on an \m

[11:21:00 root@centos8 ~]#sed -n ' ' /etc/issue
[11:21:07 root@centos8 ~]#sed 'p' /etc/issue
\S
\S
Kernel \r on an \m
Kernel \r on an \m

[11:21:16 root@centos8 ~]#sed -n 'p' /etc/issue
\S
Kernel \r on an \m

[11:21:59 root@centos8 ~]#sed -n '1p' /etc/issue
\S
[11:22:10 root@centos8 ~]#sed -n '2p' /etc/issue
Kernel \r on an \m
```



```
[11:26:51 root@centos8 ~]#ifconfig ens160| :
ens160: flags=4163<UP,BROADCAST,RUNNING,MUL
    inet 192.168.1.85 netmask 255.255.255.0
    inet 192.168.1.85 netmask 255.255.255.0
    inet6 fe80::f54:7ac3:581e:c117 pre
    inet6 2409:8a20:be61:7a20:199e:9165:0:0
    ether 00:0c:29:0b:7c:79 txqueuelen 1000
    RX packets 22528 bytes 15943619 (15.5 MiB)
    RX errors 0 dropped 1468 overruns 0 (0.0%)
    TX packets 8950 bytes 784348 (765.5 KiB)
    TX errors 0 dropped 0 overruns 0 (0.0%)
```

```
[11:27:09 root@centos8 ~]#ifconfig ens160| :
    inet 192.168.1.85 netmask 255.255.255.0
```



不显示注释和空行

```
[11:35:58 root@centos8 ~]#sed '/^#/d;/^$/d' /etc/fstab
UUID=7a405199-a904-4290-89a3-72992b4ad489 /
UUID=6c58d4aa-0c80-4e0b-b372-0a2187033d23 /l
UUID=2dd6f870-9df8-47ec-a4fc-b8a01878f4e0 /l
UUID=27f69466-124d-4124-aab8-85f247d76d57 n
[11:36:03 root@centos8 ~]#cat /etc/fstab
```

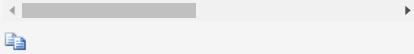
```
#
# /etc/fstab
# Created by anaconda on Tue Apr  5 08:42:30 2016
#
# Accessible filesystems, by reference, are
# See man pages fstab(5), findfs(8), mount(8)
#
# After editing this file, run 'systemctl daemon-reload'
# units generated from this file.
#
UUID=7a405199-a904-4290-89a3-72992b4ad489 /
UUID=6c58d4aa-0c80-4e0b-b372-0a2187033d23 /l
UUID=2dd6f870-9df8-47ec-a4fc-b8a01878f4e0 /l
UUID=27f69466-124d-4124-aab8-85f247d76d57 n
[11:36:15 root@centos8 ~]#
```



使用 -i 选项 直接 修改文本内容, 最好-i.bak 备份修

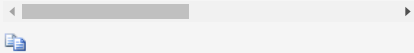
```
[11:38:40 root@centos8 ~]#sed -i '/^#/d;/^$/d' /etc/fstab
```

```
[11:38:51 root@centos8 ~]#cat /etc/fstab
UUID=7a405199-a904-4290-89a3-72992b4ad489 /
UUID=6c58d4aa-0c80-4e0b-b372-0a2187033d23 /l
UUID=2dd6f870-9df8-47ec-a4fc-b8a01878f4e0 /v
UUID=27f69466-124d-4124-aab8-85f247d76d57 n
```



示例：搜索替换和&

```
[11:39:01 root@centos8 ~]#sed -nr 's#U..D#&I
UUIDname=7a405199-a904-4290-89a3-72992b4ad4i
UUIDname=6c58d4aa-0c80-4e0b-b372-0a2187033d;
UUIDname=2dd6f870-9df8-47ec-a4fc-b8a01878f4i
UUIDname=27f69466-124d-4124-aab8-85f247d76d!
```

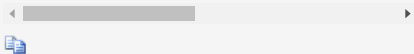


示例：去文件的前缀和后缀

```
[12:03:53 root@centos8 ~]#echo a.b.c.gz | s
a.b.c
[12:08:20 root@centos8 ~]#echo a.b.c.gz | s
gz
```

将非#开头的行加#

```
[12:09:25 root@centos8 ~]#sed -rn "s/^[^#]/;
#UUID=7a405199-a904-4290-89a3-72992b4ad489 .
#UUID=6c58d4aa-0c80-4e0b-b372-0a2187033d23 .
#UUID=2dd6f870-9df8-47ec-a4fc-b8a01878f4e0 .
#UUID=27f69466-124d-4124-aab8-85f247d76d57 I
```



3、文本处理之awk

gawk：模式扫描和处理语言，可以实现下面功能

- 文本处理

- 输出格式化的文本报表
- 执行算数运算
- 执行字符串操作

格式:

```
awk [options] 'program' var=value file...
awk [options] -f programfile var=value
```

program通常是被放在单引号中，并可以由三种部分组成

- BEGIN语句块
- 模式匹配的通用语句块
- END语句块

常见选项:


- -F “分隔符” 指明输入时用到的字段分隔符，默认的分隔符是若干个连续空白符
- -v var=value 变量赋值

动作print

格式:

```
print item1, item2, ...
```

- 逗号分隔符
- 输出item可以是字符串，也可以是数值；当前记录的字段、变量或awk的表达式
- 如省略item，相当于print \$0
- 固定字符需要用" "引起来，而变量和数字不需要



```
[root@centos8-]#df
Filesystem      1K-blocks    Used Available
devtmpfs         980100         0    980100
tmpfs            998536         0    998536
tmpfs            998536     8912    989624
tmpfs            998536         0    998536
/dev/sda2       104806400 4922212 99884188
/dev/sda5        52403200 398400 52004800
/dev/sda1         999320 177712 752796
/dev/sr0         9046654 9046654 0
tmpfs           199704         0    199704
[root@centos8-]#df |awk '{print $1,$5}'
Filesystem Use%
devtmpfs 0%
```

```
tmpfs 0%
tmpfs 1%
tmpfs 0%
/dev/sda2 5%
/dev/sda5 1%
/dev/sda1 20%
/dev/sr0 100%
tmpfs 0%
```



取 ifconfig 输出结果中的IP地址



```
[root@centos8~]$ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.88 netmask 255.255.255.0
    inet6 fe80::250:56ff:fe33:298d prefixlen 64 scopeid 0x20:::
    ether 00:50:56:33:29:8d txqueuelen 1000 (Local Loopback)
    RX packets 1624 bytes 172562 (168.1 KiB)
    RX errors 0 dropped 0 overruns 0 (0 B)
    TX packets 1419 bytes 90960 (88.8 KiB)
    TX errors 0 dropped 0 overruns 0 (0 B)

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x1:::
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 (0 B)
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 (0 B)
```

```
[root@centos8~]$ifconfig eth0 | awk '/netmask/ {print $2}'
10.0.0.88
```

```
[root@centos8~]$ifconfig eth0 | sed -rn '2s/.*addr //'
10.0.0.88
```



文件host_list.log 如下格式，请提取".magedu.com"前面的主机名部分并写入到回到该文件中



```
[root@centos8~]$cat host_list.log
```

```
1 10.0.0.88 magedu.com
```



```
1 www.magedu.com
2 blog.magedu.com
3 study.magedu.com
4 linux.magedu.com
5 python.magedu.com
```



```
[root@centos8~]$awk -F"[.]" '{print $2}' host_list.log
[root@centos8~]$cat host_list.log
1 www.magedu.com
2 blog.magedu.com
3 study.magedu.com
4 linux.magedu.com
5 python.magedu.com
www
blog
study
linux
python
```



awk变量

awk中的变量分为：内置和自定义变量

常见的内置变量

- FS: 输入字段分隔符，默认为空白字符，功能相当于-F

-F 和 FS变量功能一样，同时使用会冲突

-F 和 FS变量功能一样，同时使用会 -F 优先级高

```
awk -v FS=":" '{print $1,FS,$3}' /etc/passwd
awk -v FS=";" '{print $1FS$3}' /etc/passwd
awk -F: '{print $1,$3,$7}' /etc/passwd
```

```
S=:;awk -v FS=$S '{print $1FS$3}' /etc/passwd
```



```
[root@centos8~]$awk -v FS=":" '{print $1FS$:}' /etc/passwd
root:0
bin:1
daemon:2
adm:3
```



- OFS: 输出字段分隔符，默认为空白字符

```
[root@centos8~]$awk -v FS=":" '{print $1,$3
root 0 /bin/bash
bin 1 /sbin/nologin
daemon 2 /sbin/nologin
adm 3 /sbin/nologin
[root@centos8~]$awk -v FS=":" -v OFS=":" '{|
root:0:/bin/bash
bin:1:/sbin/nologin
daemon:2:/sbin/nologin
adm:3:/sbin/nologin
```

- RS: 输入记录record分隔符，指定输入时的换行符

```
[root@centos8~]$cat /etc/passwd | head -4
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
[root@centos8~]$awk -v RS=':' '{print }' /e
root
x
0
0
```

- ORS: 输出记录分割符，输出时用指定符号代替换行符

```
[root@centos8~]$awk -v RS=':' -v ORS='--' '.
root--x--0--0--root--/root--/bin/bash
bin--x--1--1--bin--/bin--/sbin/nologin
daemon--x--2--2--daemon--/sbin--/sbin/nolog
adm--x--3--4--adm--/var/adm--/sbin/nologin
```

- NF字段数量

应用变量时，变量前不需要加\$

```
[root@centos8~]#df
Filesystem      1K-blocks    Used Available
devtmpfs        980100         0    980100
tmpfs           998536         0    998536
tmpfs           998536     8912    989624
tmpfs           998536         0    998536
/dev/sda2      104806400 4922224 99884176
/dev/sda5       52403200  398400 52004800
/dev/sda1       999320    177712  752796
/dev/sr0        9046654 9046654      0
tmpfs          199704         0    199704
```

```
[root@centos8~]$df |awk '{print NF}'
```

7
6
6
6
6
6
6
6
6
6

取倒数第二列和最后一列

```
[root@centos8~]$df |awk '{print $(NF-1)}'
```

Mounted

0%

0%

1%

0%

5%

1%

20%

100%

0%

```
[root@centos8~]$df |awk '{print $(NF)}'
```

on

/dev

/dev/shm

/run

/sys/fs/cgroup

/

/data

/boot

/mnt/cdrom

/run/user/0



- NR: 记录的编号



```
[root@centos8~]$awk '{print NR,$0}' /etc/fs
1
2 #
3 # /etc/fstab
4 # Created by anaconda on Mon Apr 18 10:31
5 #
6 # Accessible filesystems, by reference, a
7 # See man pages fstab(5), findfs(8), moun
8 #
9 # After editing this file, run 'systemctl
10 # units generated from this file.
11 #
12 UUID=578e255d-d098-4f3c-8723-be0ec693dc8:
13 UUID=d1918dd5-8b34-47ee-8b82-bf0af3f171a:
14 UUID=0b950f5b-2bce-49a3-940b-d21f4519503l
15 UUID=03bc96b3-1bf4-4e88-be0f-48e845b8285:
16 UUID=2020-11-18-21-39-52-00 /mi
```



取ifconfig输出结果中的IP地址

```
[root@centos8~]$ifconfig eth0 | awk 'NR==2{
10.0.0.88
```



- FNR各文件分别计数，记录的编号



```
[root@centos8~]$awk '{print NR,$0}' /etc/is:
1 \S
2 Kernel \r on an \m
3
4 CentOS Linux release 8.5.2111
[root@centos8~]$awk '{print FNR,$0}' /etc/i:
1 \S
2 Kernel \r on an \m
3
4 CentOS Linux release 8.5.2111
```





- FILENAME: 当前文件名

```
[root@centos8~]$awk '{print FILENAME,$0}' .  
/etc/issue \S  
/etc/issue Kernel \r on an \m  
/etc/issue
```

- ARGV: 命令行参数的个数

```
[root@centos8~]$awk '{print ARGV}' /etc/issue  
3  
3  
3  
3  
[root@centos8~]$awk 'BEGIN{print ARGV}' /etc/issue  
3
```

- ARGV: 数组，保存的是命令行所给定的各参数，每一个参数: ARGV[0],

分类: [Linux-系统应用-命令](#)

标签: [文本处理三剑客](#)

好文要顶

已关注

收藏该文



goodbay说拜拜
粉丝 - 2 关注 - 1

我在关注他 取消关注

1

推荐

0

反对

支持成功


« 上一篇: [Linux正则表达式](#)

» 下一篇: [软件包管理 yum私有仓库](#)

posted @ 2022-04-21 15:20 [goodbay说拜拜](#) 阅读

(20) 评论(0) 编辑 收藏 举报

[刷新评论](#) [刷新页面](#) [返回顶部](#)

 发表评论

编辑

预览

B



</>



支持 Markdown

 自动补全

提交评论

退出

订阅评论

我的博客

[Ctrl+Enter快捷键提交]

【推荐】中国云计算领导者：阿里云轻量应用服务器2核2G低至108元/年

【推荐】第五届金蝶云苍穹低代码开发大赛正式启动，百万奖金等你拿！

编辑推荐：

- 如何将一个实例的内存二进制内容读出来？
- K8S | 核心原理分析
- 如何获取 C#程序 内核态线程栈
- 如何计算一个实例占用多少内存？
- [MAUI 程序设计] 用 Handler 实现自定义跨平台控件

即构专区：

- 线上狼人杀太卡？快试试这个新功能
- 即构自研海量有序数据网络MSDN，构建全球可靠的多云通讯链路
- 零基础实现Java直播（二）：实现流程
- 永远拥抱开放生态 | Metaworld2.0能力发布
- 端午“沉浸式云旅游”怎么玩？即构助力“直播+”新场景落地