awk

前置指令 | awk 选项 条件 指令

awk 选项 条件 指令 被处理的文档

-F print $0 $1 $2 NR NF

BEGIN{ }

{ } 逐行任务

END{ }

练习一：awk的条件

1. 使用正则表达式作为条件

/ / ~ 包含 !~ 不包含

awk '/root/{print}' user 找有root的行

awk -F: '$1~/root/{print}' user 找第1列包含root的行

awk '/bin/{print}' user 找有bin的行

awk -F/ '$3~/bin/{print}' user 找第3列包含bin的行

awk -F/ '$3!~/bin/{print}' user 找第3列不包含bin的行

1. 使用数字或者字符串

== != > < >= <=

awk 'NR==1{print}' user 找第1行

awk 'NR==1' user 找第1行，如果任务就是print，可以省略

awk -F: 'NR==1{print $1}' user 找第1行的第1列

awk 'NR!=2' user 输出第2行以外的行

awk -F: '$3<10' /etc/passwd 找第3列(UID)小于10的

awk -F: '$3>999' /etc/passwd 找第3列大于999的

awk -F: '$3<=10' /etc/passwd 找第3列小于等于10的

awk -F/ '$3~/bin/' user 找第3列包含bin的

awk -F/ '$3=="bin"' user 找第3列等于bin的

3， && 并且 || 或者

awk -F: 'NR>2&&NR<5' /etc/passwd 找3~4行

awk -F: 'NR>=2&&NR<=5' /etc/passwd 找2~5行

awk -F: 'NR>10&&NR<3' /etc/passwd 找大于10并且小于3的行，无输出，逻辑错误

awk -F: 'NR>10||NR<3' /etc/passwd 找大于10或者小于3的行

awk -F: '$1=="root"||$1=="bin"' /etc/passwd 找第1列是root或者第1列是bin的行

awk -F: '$1=="root"&&$1=="bin"' /etc/passwd 逻辑错误，无输出

1. 运算

awk 'BEGIN{print 1+1 }'

awk 'BEGIN{print 2-1 }'

awk 'BEGIN{print 2\*2 }'

请在1~100中找到7的倍数或者含有7的

seq 100 | awk '$1%7==0||$1~/7/'

练习二：在awk中使用if

单分支

if( ){ }

awk -F: 'BEGIN{x=0}{if($7~/bash/){x++}}END{print x}' /etc/passwd 首先定义变量x=0，然后逐行任务找passwd文档中哪行的第7列包含bash，每找到一个这样的行就把x+1，最后使用end任务输出x的最终值，相当于输出服务器中使用bash做解释器的账户数量

双分支

if( ){ }else{ }

awk -F: 'BEGIN{x=0;y=0}{if($7~/bash/){x++}else{y++}}END{print x,y}' /etc/passwd 定义2个变量，初始值都为0，然后查找passwd文档中解释器包含bash的把x+1，否则y+1，最终输出x与y的值，相当于最终输出使用bash的账户数量与不使用bash的账户数量

多分支

if( ){ }else if( ){ }else{ }

awk -F: 'BEGIN{x=0;y=0;z=0}{if($7~/bash/){x++}else if($7~/nologin/){y++}else{z++}}END{print x,y,z}' /etc/passwd

定义xyz三个变量，值为0，使用if多分支判断，找到使用bash的账户把x+1，找到使用nologin的账户把y+1，除此之外的账户把z+1，最后输出

练习三：在awk中使用数组

1，数组 可以当成能存多个值的特殊变量

数组名称[下标1]=值1

数组名称[下标2]=值2

awk 'BEGIN{a[1]=10;a[2]=20;print a[1]}' 定义了数组a ，然后分别定义了2个下标以及对应的值， 最后输出数组a下标是1的值

awk 'BEGIN{a[1]=10;a[2]=20;print a[2]}'

awk 'BEGIN{a["abc"]="abcabc";print a["abc"]}' 数组的下标与值可以不是数字，但使用时要加双引号

2，使用for循环遍历数组

for(变量名 in 数组名){print 变量名 }

awk 'BEGIN{a[1]=100;a[2]=200;a[3]=300;for(i in a){print i}}' 先定义了数组a，分别有3个下标以及3个值，之后使用for循环输出数组a的所有下标

awk 'BEGIN{a[1]=100;a[2]=200;a[3]=300;for(i in a){print a[i]}}' 使用for循环输出数组a的所有值

3，使用数组加for循环分析abc文件中不同的内容有什么，分别有几行

首先创建素材

[root@svr7 opt]# cat abc

abc

xyz

abc

opq

xyz

abc

然后使用

awk '{a[$1]++}END{for(i in a){print i,a[i]}}' abc

首先{a[$1]++}可以按照以下顺序对abc文件进行分析，从中获取数组的不同下标以及对应的值，最终在使用for循环输出所有。

a[abc]++ a[abc]=1

a[xyz]++ a[xyz]=1

a[abc]++ a[abc]=2

a[opq]++ a[opq]=1

a[xyz]++ a[xyz]=2

a[abc]++ a[abc]=3

1. 使用awk统计网站访问量

yum -y install httpd 安装网站服务

systemctl restart httpd 开启网站服务

curl 192.168.4.7 使用字符的方式访问网站，另外用真机的浏览器也可以

awk '{ip[$1]++}END{for(i in ip){print ip[i],i} }' /var/log/httpd/access\_log 然后对网站的日志做数据分析，得出哪个用户（ip地址），从该网站获取了多少次数据（该数据和访问量成正比）

awk '{ip[$1]++}END{for(i in ip){print ip[i],i} }' /var/log/httpd/access\_log | sort -rn 使用sort可以再最后进行排序，访问量最大的地址显示在最上面，-r是降序排列，不加就是访问量最小的地址显示在最上面，-n是以数字进行排序

练习四：编写收集服务器信息的脚本

#!/bin/bash

while :

do

uptime | awk '{print "15分钟内主机cpu的负载是"$NF}'

ifconfig eth0 | awk '/RX p/{print "eth0网卡的接收流量是"$5"字节"}'

free -h | awk '/^Mem/{print "主机内存剩余容量是"$4}'

df -h | awk '/\/$/{print "硬盘根分区剩余空间是"$4}'

awk 'END{print "主机中拥有的账户数量是"NR"个"}' /etc/passwd

echo "当前登录服务器的账户数量是 $(who | wc -l)个"

echo "当前主机运行的进程数量是$(ps aux | wc -l)个"

echo "当前主机安装的软件包数量是$(rpm -qa | wc -l)个"

sleep 3

clear

done

练习五：编写脚本，每2分钟检查是否有人尝试登陆服务器，如果尝试次数大于2次，就发邮件给管理员

#!/bin/bash

x=$(awk '$0~/Failed/&&$0!~/invalid/{ip[$11]++}END{for(i in ip){print i,ip[i]}}' /var/log/secure | awk '$2>2')

[ -z "$x" ] && exit

echo "有人试图入侵服务器，相关信息是$x" | mail -s test root

chmod +x test02.sh 脚本写完后要添加x权限

crontab -e 编写计划任务

\*/2 \* \* \* \* /opt/test02.sh

运维课程环境要求

要配置好ip地址、yum仓库、主机名、同网段主机可以ping通，带图形化安装，或者最小化安装均可，1颗cpu，1G内存即可

proxy 192.168.4.5 192.168.2.5

web1 192.168.2.100

web2 192.168.2.200

client 192.168.4.10

配置视频：

链接：https://pan.baidu.com/s/1vEFmeC3qlv2PeGz4MWSD7Q

提取码：b3pw