**AWK**

awk 编程语言/数据处理引擎

1. 基本操作方法

格式：awk [选项] ‘[条件]{指令}’ 文件

$0代表当前行全部内容

awk -F"[,:]" '{print $3}' qq.txt //以“,”或“:“ 作为分隔符打印第3列

awk -F: '{print $1}' /etc/passwd //以“:” 为分隔符打印第1列

NF：当前处理行的字段数 (列数) N：now F： 列

NR：记录当前以读入行的数量 (行数) N：now R：row

awk -F: ‘{print $NF}’ /etc/passwd //打印每一行的最后一列

awk -F: '{print $1,$7}' /etc/passwd //“,”打印出来以空格显示

df -h | awk '/\/$/{print $4}' //打印以”/”结尾的行的第四列,根分区的可用容量

awk '/Accepted/{print $11}' /var/log/secure

AWK **BEGIN** and **END**

awk会逐行处理文本，支持在处理第一行之前做一些准备工作，以及在处理完最后一行之后做一些总结性质的工作。条件加在指令”{}”前，可以使用正则，BEGIN和END可以单独只使用一个

格式：

awk [选项] ' BEGIN{指令} {指令} END{指令}' 文件

A. BEGIN{ } 行前处理，读取文件内容前执行，指令执行1次

B. { } 逐行处理，读取文件过程中执行，指令执行n次

C. END{ } 行后处理，读取文件结束后执行，指令执行1次

awk 'BEGIN{x=1.55;y=4.33;print x+y}' //可以作计算器使用，BIGIN{}在读文件前执行

awk 'BEGIN{x=0}/bash$/{x++}END{echo $x}' /etc/passwd

//统计以bash结尾的行数，统计shell为bash的用户个数

awk -F: 'BEGIN{print "用户名 UID 家目录";x=0}

{print $1,$3,$6;x++}END{print "总用户:"x}' /etc/passwd | column -t

// 用户名 UID 家目录

...... xx /.....

总用户：xx

column -t //自动用tab键对齐

**AWK条件处理**

指令可以忽略，作用与grep相同，条件使用正则

支持正则和扩展正则

1）支持对任何地方的条件作正则匹配

awk -F: '$1~/root/' /etc/passwd “~”作正则匹配的标示符 红色表示正则 紫色表示列

**awk -F: '$1!~/root/' /etc/passwd “!” 将后面的正则条件过滤，不匹配**

awk -F: '$7!~/nologin$/{print $1,$7}' /etc/passwd

//输出其中登录Shell不以nologin结尾（对第7个字段做!~反向匹配）的用户名、登录Shell信息

1. 使用数值/字符串比较设置条件

比较符号：==(等于) !=（不等于） >（大于）

>=（大于等于） <（小于） <=（小于等于）

使用 NR当前号 NF当前行的列数

**使用数值**

**Seq 5 //产生1~5的序列**

awk 'NR==3' /etc/passwd //输出第三行的用户记录

awk -F: '$3>=1000{print $1}' /etc/passwd //打印账户UID大于等于

1000的账户名称 红色表示条件

运算：

**字符串比较**

awk -F: '$1=="root"{print $1}' /etc/passwd //打印root的用户名

awk -F: '$1=="root"' /etc/passwd //打印第一列为root的行 上下对照

**逻辑测试条件**

awk -F: '$3<=1000&&$3>900' /etc/passwd //打印UID在900-100的行

**变量/常量**

未定义的变量其值默认为0，可以直接使用

awk 'BEGIN{x++;print x}'

seq 50 | awk '$1%2==0{print $1}' //找出50以内的偶数

seq 50 | awk '$1%2==0' //没有指令默认输出符合条件的一整行内容

AWK流程控制

**AWK if语句：**

if(条件){命令}else{命令}

if语句放在指令的{}里面

awk -F: '{if($3>=1000){x++}else{y++}}END{print "普通用户："x"\n系统用户："y}' /etc/passwd

//统计普通用户和系统用户的数量 \n换行

**AWK 数组**

在awk的指令{}里使用

数组元素默认值为零

数组是一个可以存储多个值的变量，具体使用的格式如下：

定义数组的格式：数组名[下标]=元素值

调用数组的格式：数组名[下标]

遍历数组的用法：for(变量 in 数组名){print 数组名[变量]}

awk 'BEGIN{a[0]="xiao";a[1]=4;print a[0],a[1]}'

结果：xiao 4

awk 'BEGIN{a[0]="xiao";a[1]=4;for(i in a){print a[i]}}'

结果：xiao 4 //for循环遍历,for中i表示下标

注意，awk数组的下标除了可以使用数字，也可以使用字符串，字符串需要使用双引号

awk 'BEGIN{a["hehe"]=11;print a["hehe"]}'

结果： 11

//使用ab模拟器让100个client去访问192.168.4.207的web服务10万次

ab -c 100 -n 100000 <http://192.168.4.207/>

ip --- [时间] get /index.html

DOS deny of service //拒绝服务攻击

awk '$0!~/Brown/' data.txt //不写指令，默认打印所有一行的所有列，即打印

所有，这里的“$0”表示整行，“!~/Brown/”表示 不匹配，该命令表示打印不包含“Brown”的行