**Morning：**

1. awk工具

-概述：swk是一门编程语言，主要学习数据处理引擎。用于查找非编辑。

《sed & awk》page 600.

-原理：逐行读取-->判断条件-->打印。

-作用：数据截取、统计

2、awk基本操作

1）基本语法格式：awk [选项] '[条件]{指令}' 文件

（若未指定分隔符，则默认将空格、制表符等作为分隔符）

举例：awk '{print $1,$3}' test.txt #打印文档1,3列

df -h | awk '{print $4}' #打印磁盘的剩余空间

1. 选项：

-F指定分隔符

awk -F: ‘{print $1}’ a.txt #指定:为分隔符

awk -F[:\\t] ‘{print $1}’ a.txt #指定:或tab为分隔符

1. 常用内置常量：

$0 文本当前行的全部内容

$3 文本的第3列

NR 文件当前行的行号

NF 文件当前行的列数（有几列）

Print不加,号：输出结果没有空格隔开

如：awk ‘{print $(NF-1)}’ passwd.txt #打印倒数第2列

awk -F: '{print NR,NF}' passwd.txt

输出： 1 7

.. ..

print指令打印常量：

如：awk -F: '{print $1,"的解释器:",$7}' /etc/passwd

输出： root 的解释器: /bin/bash

... ...

1. 案例：

-提取本机的网络流量：

ifconfig eth0 | awk '/RX p/{print “接收流量：”$5}'

-提取根分区剩余容量：

df | awk '/\/$/{print $4}'

-获取远程失败的IP地址：

tailf /var/log/secure #动态看文件的后10行，查看文本规律

awk '/Failed/{print $1,$2,$3,$11}' /var/log/secure

1. awk的处理时机（初始化与总结）
2. 语法格式：

awk [选项] ' BEGIN{指令} {指令} END{指令}' 文件

-BEGIN{ }，整个文件读取开始前执行，NR=0，指令执行1次

-{ } 逐行处理，读取文件过程中执行，指令执行n次

-END{ }，整个文件读取结束后执行，NR=最后一行，指令执行1次

1. 案例：

-统计系统中使用bash作为登录Shell的用户总个数

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

-格式化输出/etc/passwd文件的User、UID、Home

awk -F: 'BEGIN{print "User UID Home"}{print $1,$3,$NF} END{print "Total\_User："NR}' /etc/passwd | column -t #列对齐

**Afternoon：**

1. awk条件设置

1）正则（默认支持扩展正则）：

awk ‘/正则/’ a.txt #不写{}默认为{print $0}

如：awk -F: '/root/' /etc/passwd #输出包含root的行

awk -F: '$1~/root/' /etc/passwd #输出第1列包含root的行

awk -F: '$7!~/nologin$/' /etc/passwd #反向匹配

1. 数值/字符串比较：

比较符号：==、!=、>、>=、<、<=

awk -F: 'NR==3{print}' /etc/passwd #输出第3行

awk -F: '$1==”root”' /etc/passwd #输出用户名为root的行

常量比较时，必须加””，否则识别为变量

1. 逻辑测试条件：

测试符号：并&& 或||

awk -F: '$3>1 && $3<10' /etc/passwd#输出1<UID<10的行

awk -F: '$3==1 || $3==10' /etc/passwd#输出UID1或10的行

1. 数学运算

awk 'BEGIN{x++;print x}' #输出1（允许变量未定义,初始为0）

awk 'BEGIN{x=8;print x+=2}' #输出10

seq 200 | awk '$1%3==0' #找200以内3的倍数

1. 案例：

-列出100以内整数中7的倍数或是含7的数

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

-拓展：引用外部变量

awk -F: '$1=="'"$name"'"{print $1}' /etc/shadow

1. awk流程控制
2. if分支结构（常用于统计）

-语法：{if结构 指令}，指令也可以放在if结构里面。

-案例1：统计/etc/passwd文件中登录Shell是“/bin/bash”的用户个数

awk -F: '{if($7~/bash$/){i++}}END{print i}' /etc/passwd

-案例2：分别统计/etc/passwd文件中UID小于等于10、大于10的用户个数

awk -F: '{

if($3<=10)

{i++}

Else

{j++}

}

END{print i,j}' /etc/passwd

1. awk数组：

-格式：

定义数组：数组名[下标]=元素值

调用数组：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]}

注意，awk数组的下标可以使用字符串，字符串需要使用双引号，如： awk 'BEGIN{a["hehe"]=11;print a["hehe"]}'

案例：

awk 'BEGIN{a[0]++;print a[0]}' #显示1

awk 'BEGIN{a[0]=0;a[1]=11;a[2]=22; for(i in a){print i,a[i]}}'

#对a的下标进行循环，显示数组下标及所有值（下标可以是字符串）

1. awk拓展应用（用于某类数据出现多少次）

**案例：awk统计Web访问排名**

要求：分析Web日志的访问量排名，要求获得客户机的地址、访问次数，并且按照访问次数排名

awk '{IP[$1]++}END{for(i in IP){print IP[i],i}}' /var/log/httpd/access\_log | sort -nr

Tips： 1）ab -c 100 -n 100000 <http://172.25.0.11/>

#模拟100个人同时访问，点击网站10万次

1. web的日志文件放在/var/log/httpd/access\_log | error\_log
2. sort按列数对比，默认按字符串对比、升序。

-n为按数字对比，-v为降序

同理：who | awk ‘{arr[$1]++}END{for(i in arr){print i,arr[i]}}’

#统计用户登陆的数量