shell day 06

================================================

awk 选项 条件 指令

print -F

$1 $2 $3 ….. NR NF $0

BEGIN{ }

{ }

END{ }

awk中的条件

1. 使用正则作为条件

/ ^ / / $/ ~包含 !~不包含

awk -F: '/bin/{print}' user //输出有bin的行,无论bin在第几列

awk -F: '$6~/bin/{print}' user //输出第6列包含bin的行

awk -F: '$6!~/bin/{print}' user //输出第6列不包含bin的行

awk -F: '/bin/' user //以上指令还可以这样输入，使用awk时

如果指令就是{print}，且前面写了条件的情况下可以省略{print}不写

awk -F: '$6~/bin/' user

awk -F: '$6!~/bin/' user

2，使用数字和字串作为条件

==相等 !=不相等 >大于 >=大于等于 <小于 <=小于等于

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

awk -F: 'NR<3{print}' /etc/passwd //输出第1~2行

awk -F: '$3<10{print}' /etc/passwd //输出id号是0~9的行

awk -F: '$3>=1000{print}' /etc/passwd //输出普通用户的行

awk -F: '$7=="/bin/bash"{print}' /etc/passwd //输出解释器是/bin/bash的

行,注意如果对比内容不是变量,要用双引号(数字不用)

awk -F: '$7!="/bin/bash"{print}' /etc/passwd //上述情况的取反搜索

3，逻辑组合 &&并且 ||或者

awk -F: '$3>=10&&$3<=20' /etc/passwd //找id号范围是10~20的行

awk -F: 'NR>=2&&NR<=10' /etc/passwd //找2~10行

awk -F: '$3<5||$3>1000' /etc/passwd //找id号是0~4或者1001以上的行

awk 'NR<5||NR>10{print NR}' /etc/passwd //找1~4行或者11以上的行输出

行号

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

awk -F: 'NR>5||NR<100' /etc/passwd //找所有行

awk -F: 'NR<5&&NR>100' /etc/passwd //逻辑错误，没有输出

4，使用运算作为条件

awk 'BEGIN{print 1+1}' //常规运算

awk 'BEGIN{print 2-1}'

awk 'BEGIN{print 2\*2}'

awk 'BEGIN{print 2/2}'

awk 'BEGIN{print 2%2}'

seq 200 | awk '$1%7==0' //找到能被7整除的行

seq 200 | awk '$1~/7/||$1%7==0' //找被7整除或包含7的行

seq 200 | awk '$1~/7/&&$1%7==0' //找被7整除并且包含7的行

在awk中使用if

**单分支，如果满足条件，就执行指令，不满足就不执行任何指令**

awk '{if(条件){指令}}' //基本格式

awk '{if( ){ }}' //使用时可以先写框架

awk -F: '{if($7~/bash/){print}}' user //逐行检查user文档，如果

第7列包含bash，就输出该行

下课休息 11:20回

双分支

多分支