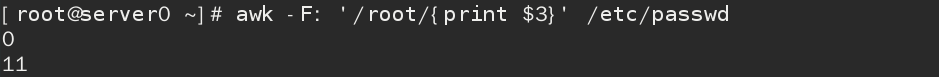
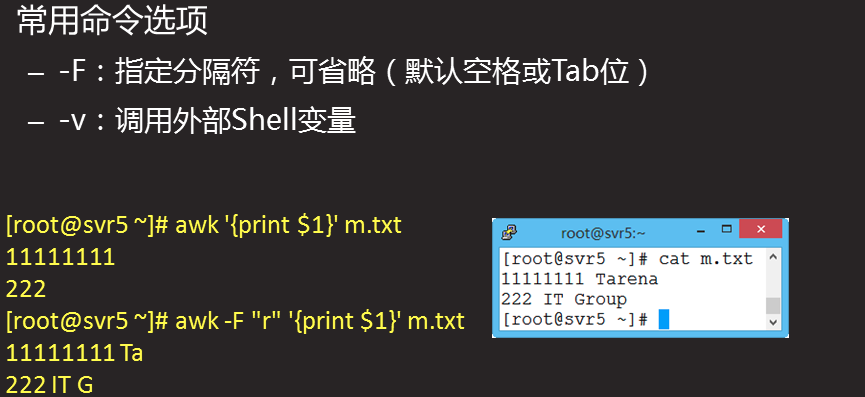
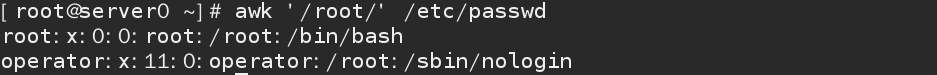
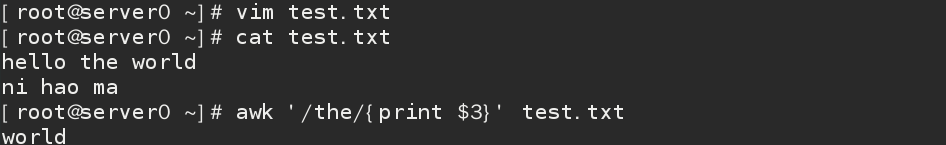
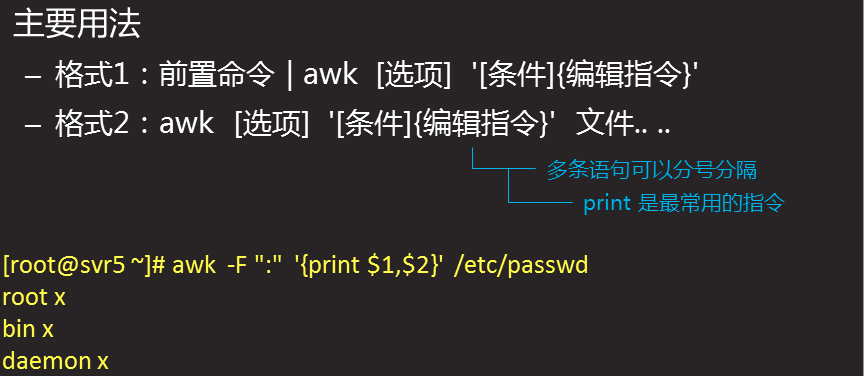
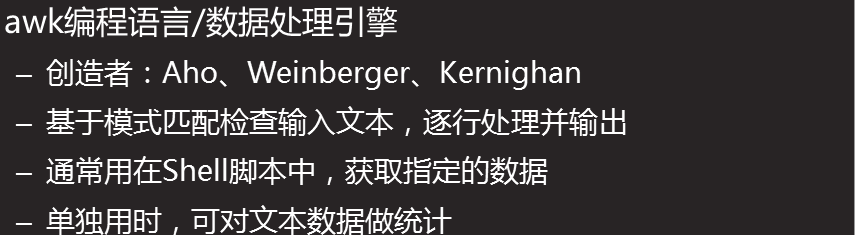
Awk 数据过滤、统计软件[=grep增强版] 逐行处理



**#也可以 -F “[:,]”**

**#条件：/正则/，默认所有**

**#指令：print，默认打印**

**#awk支持列、行,以空格或Tab键来区分列**

**#条件：/正则/，默认所有**

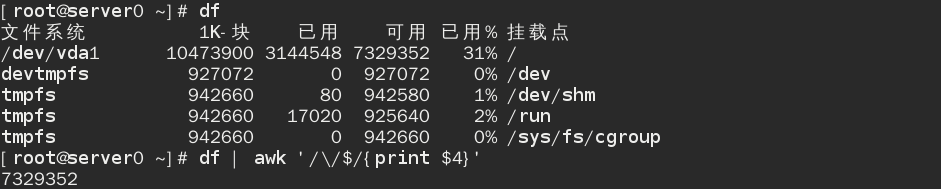
**#指令：print，默认打印**

脚本：性能分析

过滤剩余内容的值（free）



过滤根分区的剩余容量（df）



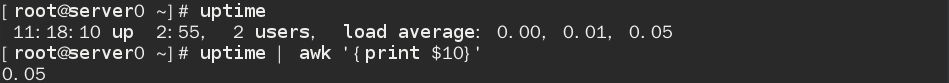
过滤本机的IP地址是什么（ifconfig）

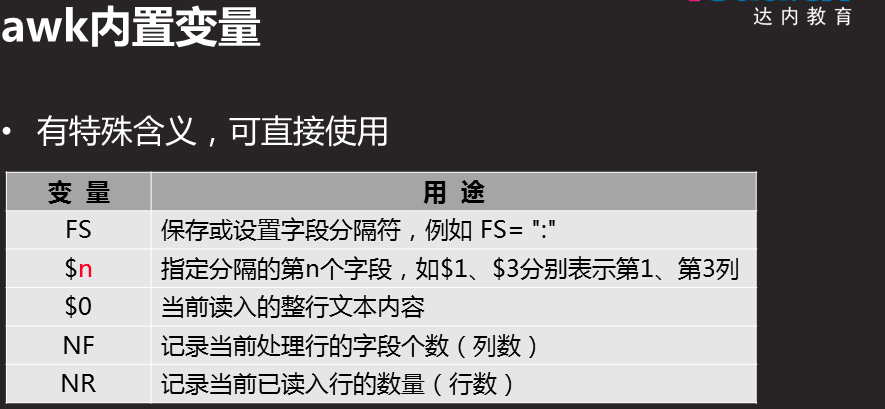


过滤eth0网卡的流量是多少（ifconfig）



过滤cpu的负载是多少（uptime或者top）



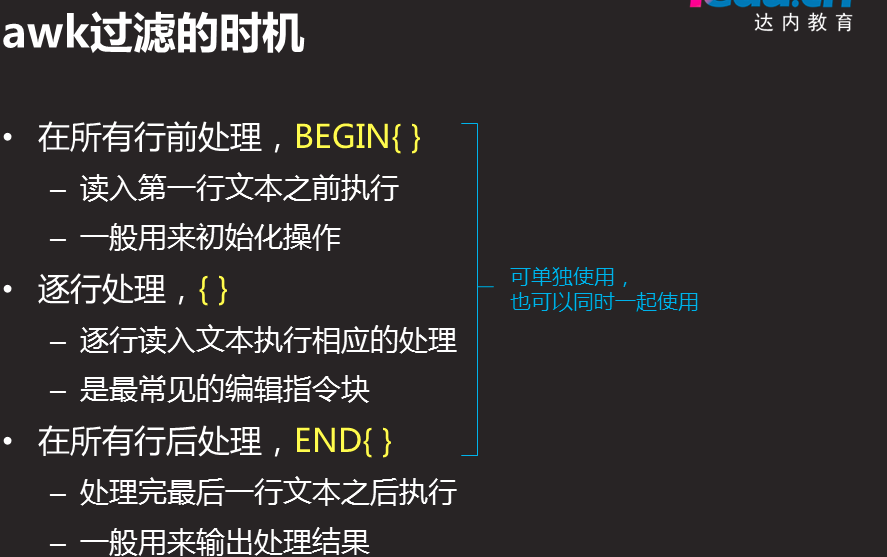


**#awk是可以打印长量的，但是常量需要用“”括起来**

**#当前行的列号**

**#当前行的行号**

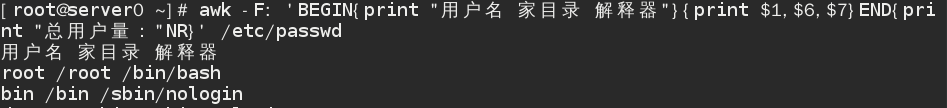
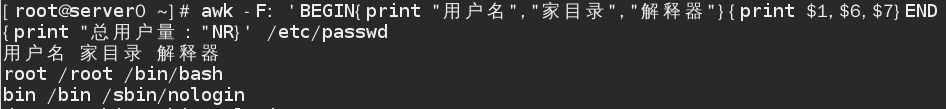


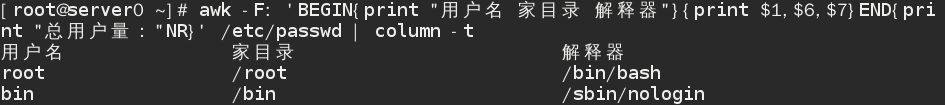


**#awk [选项] ‘BEGIN{} 条件{} END{}’ 文件**

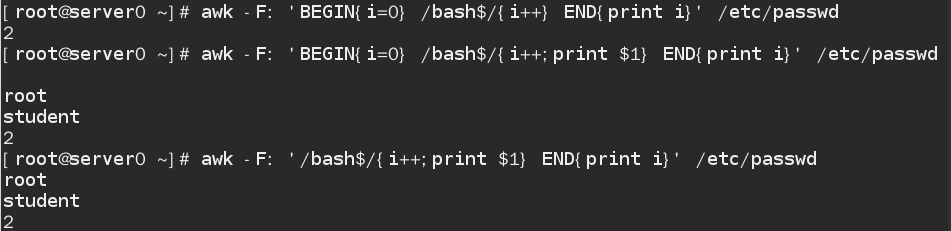
**执行1次**

**#所有的指令放在{}**





**自动对齐**

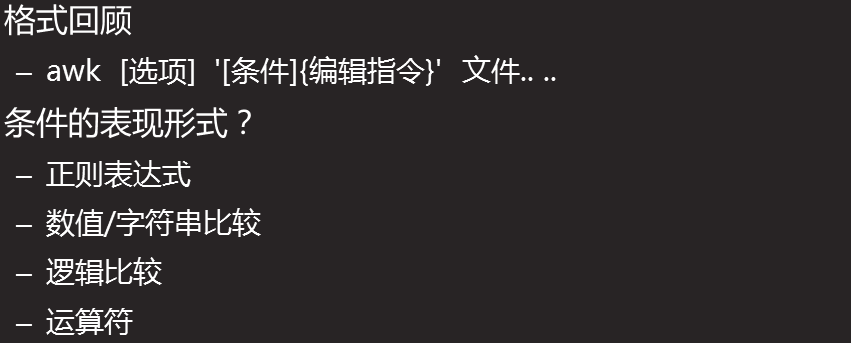


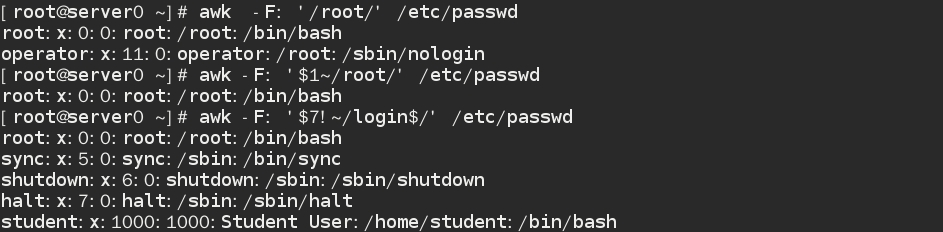
**awk 中，变量可以不定义直接使用，默认初始值是0**

从/var/log/secure中把失败的连接记录中，IP地址提取出来



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*awk处理条件\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***



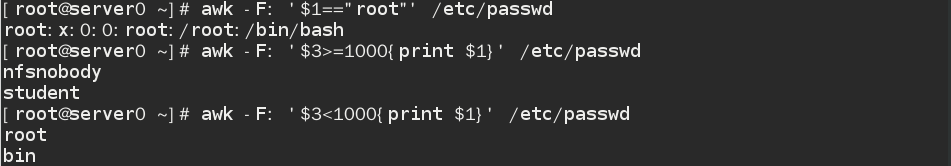


1. **正则表达式**

**模糊匹配**

**~ 匹配**

**!~ 不匹配**



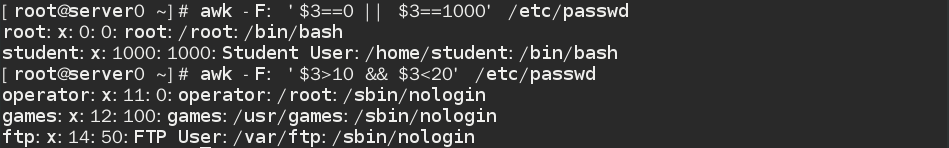
**2.数值/字符串比较 精准匹配**

**== != > <**

**>= <=**

**匹配系统用户:UID<1000**

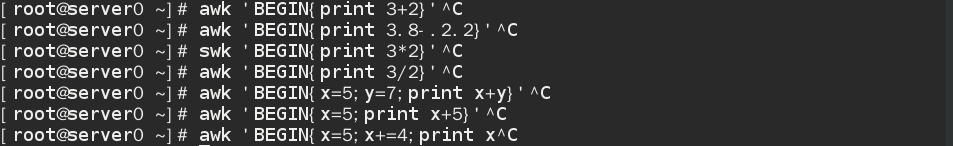
**匹配普通用户:UID>=1000**



1. **逻辑比较**

**&& 并且**

**|| 或者**

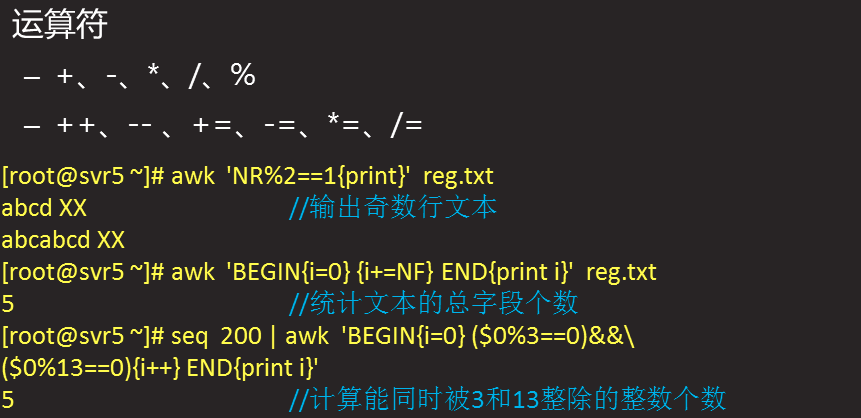
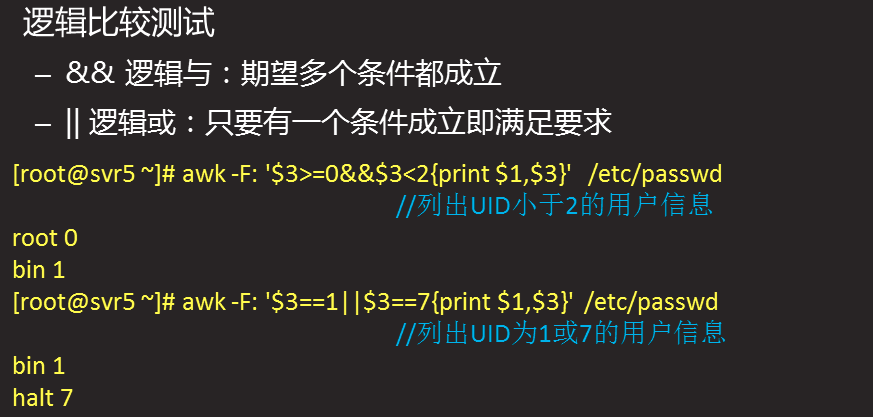
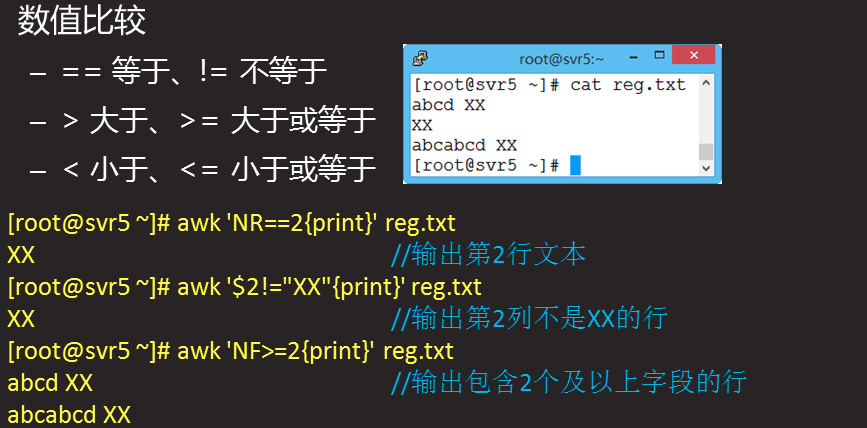
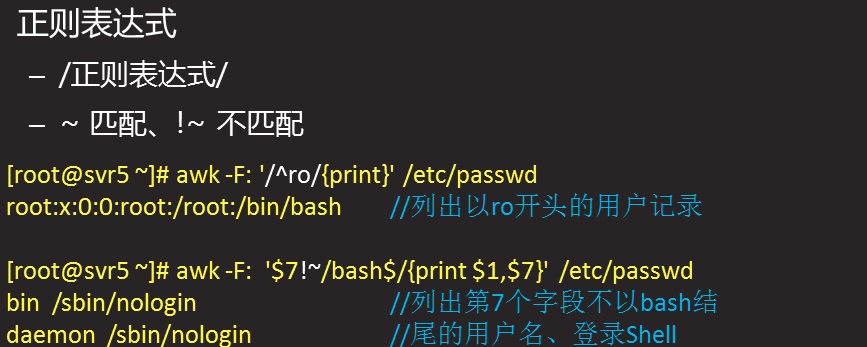
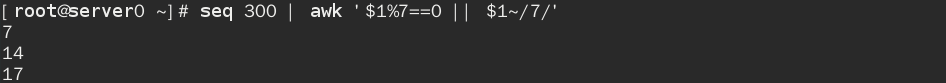


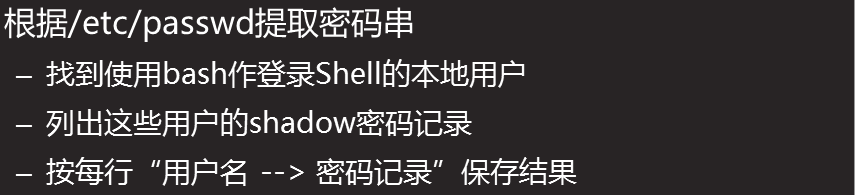
1. **算术运算**

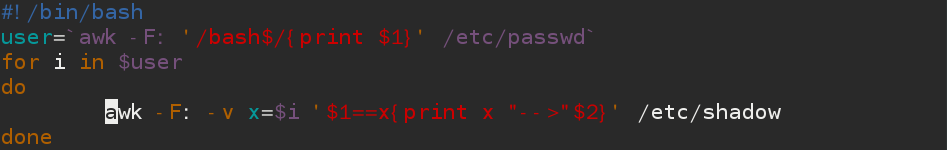
**所有运算都可以用awk运算**

题目：seq 300 | awk

把300以内，能被7整除或者包含7的数打印出来

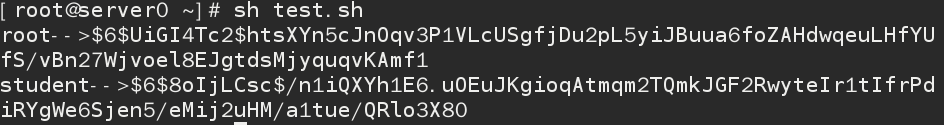


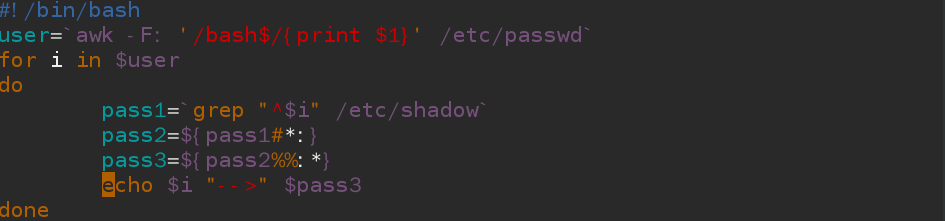




**#awk本身调用不了外部for循环的变量i，通过-v来在awk内定义一个中间变量x，这样就可以了**

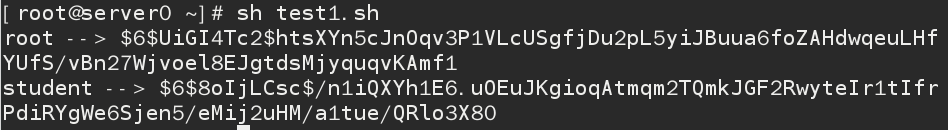
**#方法1**



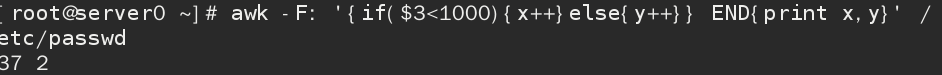
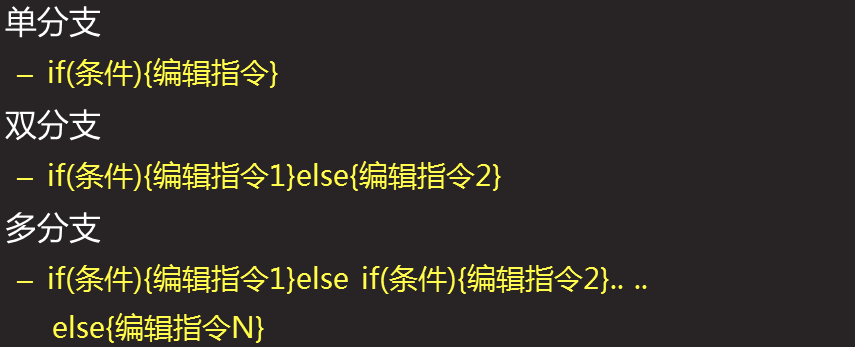


**#方法2**

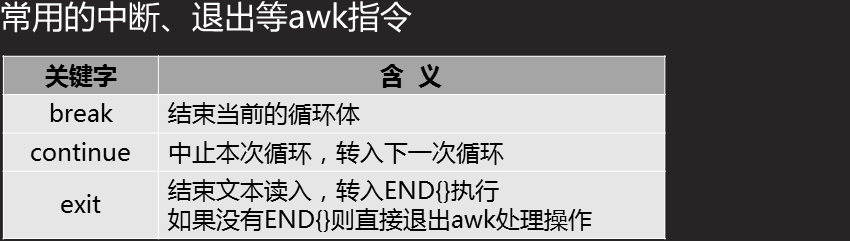
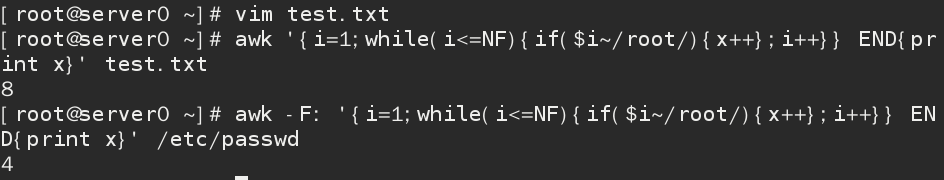
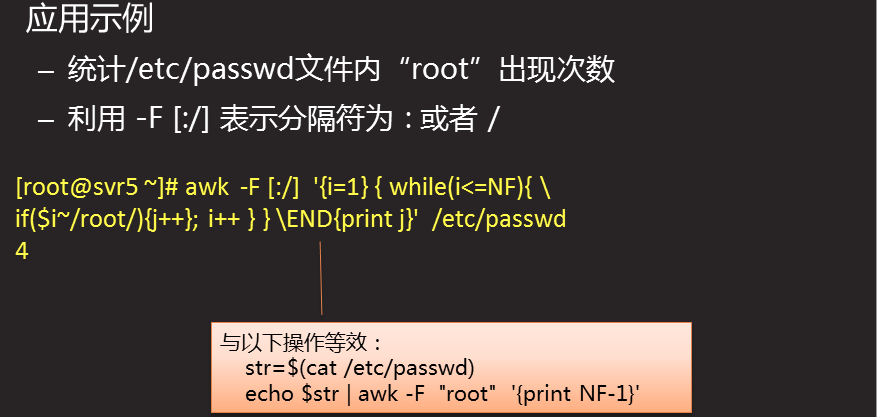
**#掐头去尾**



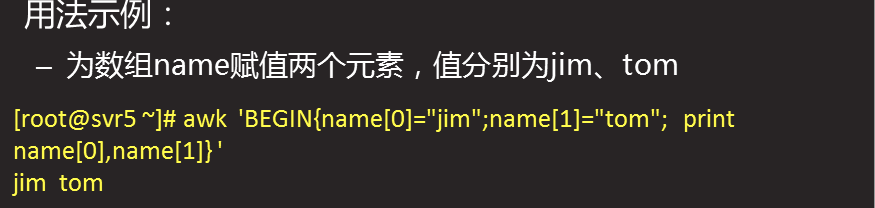
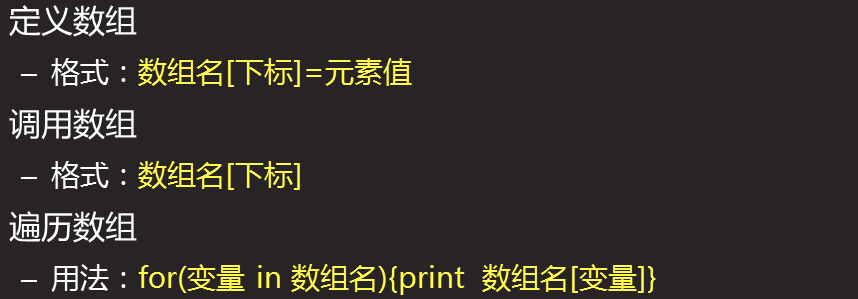
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*awk支持for,if，while\*\*\*\*\*\*\*\*\*\*\*\*\*\***



**求词频（某一个词出现的频率）**



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**用ab软件快速模拟100个人访问网站10000次（拒绝服务攻击）**

**#数组的定义及使用**



