

一、awk 流程控制

P1 if 语句

单分支 if 判断

单分支: `awk '{ if() {指令} }' 文件` `#if(),()` 里面写条件, 条件满足, 执行{指令}, 不满足则不执行

统计/etc/passwd 文件中 UID 大于等于 1000 的用户个数

```
[root@svr7 ~]# awk -F: '{ if($3>=1000){i++} } END{print i}' /etc/passwd
```

统计/etc/passwd 文件中用户名为 root 的用户名和 uid

```
[root@svr7 ~]# awk -F: '{ if($1=="root"){print $1,$3} }' /etc/passwd
```

查看 CPU 负载, 15 分钟负载是否大于等于 0.01, 大于则输出 CPU 负载:

```
[root@svr7 ~]# uptime | awk '{ if($NF>=0.01){print "CPU 的负载为: "$NF} }'
```

双分支 if 判断

双分支: `awk '{ if() {指令} else{ } }' 文件`

统计/etc/passwd 文件, 普通用户 (UID 大于等于 1000) 和系统用户 (小于 1000)

```
[root@svr7 ~]# awk -F: '{ if($3>=1000){i++}else{j++} } END{print "普通用户: "i, "系统用户: "j}' /etc/passwd
```

统计/etc 目录有多少普通文件和目录

```
[root@svr7 ~]# ls -l /etc/ | awk '{ if($1~/^-/){x++}else{y++} } END{print "普通文件: "x,"目录个数: "y}'
```

多分支 if 判断

多分支: `awk '{ if() {指令} else if(){...} ... else{ } }' 文件`

```
[root@svr7 ~]# ls -l /etc/ | awk '{ if($1~/^-/){x++}else if($1~/^d/){y++}else{z++} } END{print "普通文件: "x,"目录个数: "y,"其他: "z}'
```

P2 for 循环

for 循环

for 循环格式: `for (表达式 1;表达式 2;表达式 3) {指令}` `#表达式 1 (初始值) ;表达式 2 (条件) ;表达式 3 (步长)`

```
[root@svr7 ~]# awk 'BEGIN{ for (i=1;i<=5;i++){print i}}'      #屏幕打印 12345
```

```
[root@svr7 ~]# awk 'BEGIN{ for (i=5;i>=1;i--){print i}}'      #屏幕打印 54321
```

二、awk 数组与应用案例

P1 数组

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]}

案例：使用数组打印 jim 和 tom

```
[root@svr7 ~]# awk 'BEGIN{name[0]="jim";name[1]="tom";print name[1],name[0]}'
```

```
[root@svr7 ~]# awk 'BEGIN{name[0]="jim";name[1]="tom";print name[1],name[1],name[0]}'
```

定义的数组下标也可以不是数字，是字符串，但是要用引号引起来

```
[root@svr7 ~]# awk 'BEGIN{age["tom"]=22;age["jim"]=18;print age["jim"],age["tom"]}'
```

遍历数组的值，一次性打印数组的值

```
[root@svr7 ~]# awk 'BEGIN{x[0]=0;x[1]=1;x[2]=2;x[3]=3;x[4]=44; for(i in x){print i,x[i]} }' #每循环一次取数组的下标，但是取的时候是无序的，打印下标以及下标的值，i 调用的是下标的值
```

P2 应用案例

统计 web 访问日志

访问日志的格式

```
[root@svr5 ~]# tail /var/log/httpd/access_log
```

```
172.40.1.18-- [11/Nov/2019:11:54:22 +0800] "GET /favicon.ico HTTP/1.1" 404
```

```
209 "http://172.40.50.11 8/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
```

```
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.90 Safari/537.36"
```

简化数据，统计每个 IP 访问的次数

```
[root@svr7 ~]# vim access.log
```

```
172.40.1.18
```

```
172.40.1.17
```

```
172.40.1.18
```

```
172.40.1.16
```

```
172.40.1.17
```

```
172.40.1.18
```

```
[root@svr7 ~]# awk '{IP[$1]++} END{ for(i in IP) {print i,IP[i]} }' access.log # {IP[$1]++} 定义一个数组变量为 IP，下标为 $1，即使用文件的第 1 列作为下标
```

```
172.40.1.16 1
```

```
172.40.1.17 2
```

```
172.40.1.18 3
```

统计用户登录的次数

```
[root@svr7 ~]# who
```

#查看登录信息

```
[root@svr7 ~]# who | awk '{IP[$1]++} END{ for(i in IP) {print i,IP[i]} }'
```

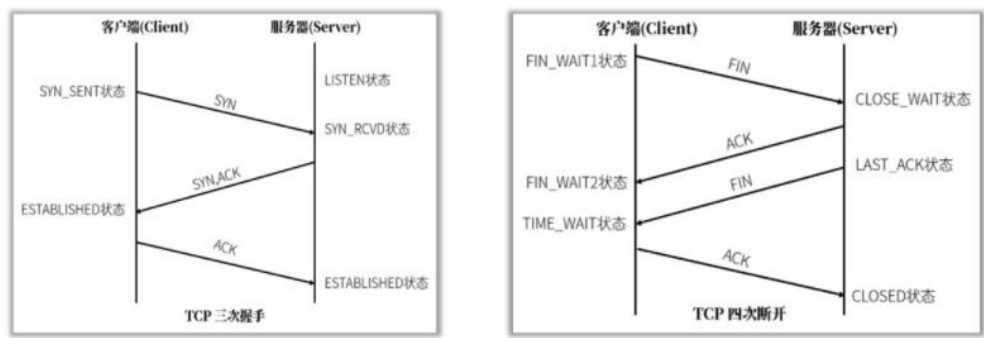
```
root 4
```

三、awk 综合案例（练习题）

P1 案例集

TCP 链接状态

- 三次握手、四次断开



ss 命令选项及描述

命令选项	功能描述
-H	不显示标题行
-n	以数字格式显示信息，不尝试解析服务名称
-a	显示所有
-p	显示进程信息
-t	显示TCP连接的信息
-u	显示UDP连接的信息
-s	显示汇总信息

```
[root@svr7 ~]# ss -at #不显示服务端口号
[root@svr7 ~]# ss -atn #显示服务端口号
[root@svr7 ~]# ss -atnp #显示进程信息
[root@svr7 ~]# ss -atnH #不显示标题行
[root@svr7 ~]# ss -s #显示汇总信息
```

P2 综合案例

监控网络连接状态脚本

```
[root@svr7 ~]# mkdir /root/shell/day05
[root@svr7 ~]# cd /root/shell/day05
[root@svr7 day05]# vim net.sh
#!/bin/bash
#所有 TCP 连接的个数
tcp_total=$(ss -s | awk '$1=="TCP"{print $2}')
#所有 UDP 连接的个数
udp_total=$(ss -s | awk '$1=="UDP"{print $2}')
#所有处于 Listen 监听状态的 TCP 端口个数（服务已经起来，但是还没有人连接）
```

```
listen_total=$(ss -antlp | awk 'BEGIN{x=0} /^LISTEN/{x++} END{print x}')  
#所有处于 ESTABLISHED 状态的 TCP 连接个数（建立连接的状态）  
estab_total=$(ss -antpH | awk 'BEGIN{count=0} /^ESTAB/{count++} END{print count}')
```

#所有处于 TIME-WAIT 状态的 TCP 连接个数（正在断，但是还没有断开的连接）

```
time_wait_total=$(ss -antpH | awk 'BEGIN{count=0} /^TIME-WAIT/{count++} END{print count}')
```

```
echo "计算机中所有处于 TCP 连接个数为: $tcp_total"  
echo "计算机中所有处于 UDP 连接个数为: $udp_total"  
echo "计算机中所有处于 listen 连接个数为: $listen_total"  
echo "计算机中所有处于 estab 连接个数为: $estab_total"  
echo "计算机中所有处于 time_wait 连接个数为: $time_wait_total"
```

```
[root@svr7 day05]# chmod +x net.sh
```

```
[root@svr7 day05]# ./net.sh
```