### 1、进程与线程

#### 区别

* 进程是系统资源分配的最小单位；线程是程序执行的最小单位。
* 进程使用独立的数据空间；线程共享进程的数据空间。

#### 线程调度（待完成）

常见线程调度算法：

轮转调度

先来先服务调度

优先级调度

多级反馈队列调度

高响应比优先调度

<https://blog.csdn.net/windyblankboy/article/details/51636931>

#### 线程切换步骤

线程的上下文切换

线程切换的代价

#### Linux下的IPC（进程间通信）——面试中间件研发的职位

需要了解以下6中进程间通信方式的原理和适用场景：

进程间数据共享的场景：共享内存

进程间数据交换的场景：UnixSocket、消息队列

Pipe

Signal

Semaphore

#### 协程（Coroutines）

比线程更加轻量级。正如一个进程可以拥有多个线程，一个线程也可以拥有多个协程。



协程不是被操作系统内核所管理，而完全是由程序所控制（也就是在用户态执行）。

优点：性能提升，不会像线程切换那样消耗资源。

Java协程框架：Kilim、Quasar

Linux命令

### 2、死锁

概念：多个进程争夺资源过程中形成的一种僵局状态。

产生原因：争夺资源；进程间推进顺序非法。

死锁产生的4个必要条件：

* 互斥条件：一个资源每次只能被一个进程使用。
* 请求与保持条件：一个进程因请求资源而阻塞时，对已获得的资源保持不放。
* 不剥夺条件：进程已获得的资源，在未使用完之前，不能强行剥夺。
* 循环等待条件：若干进程之间形成一种头尾相接的循环等待资源关系。

死锁的处理策略：预防（破坏必要条件）、避免（使用算法）、检测和解除。

### 3、Linux常用命令

#### awk

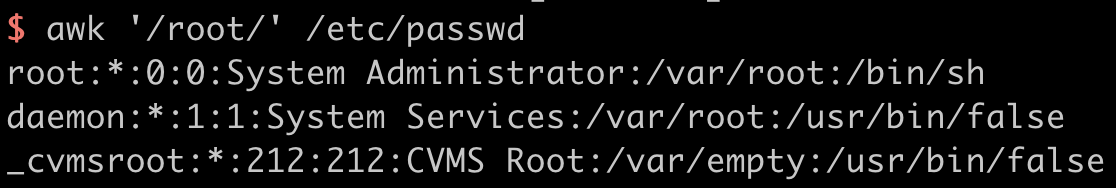
一个强大的文本分析工具，相对于grep的查找，sed的编辑，awk在其对数据分析并生成报告时，显得尤为强大。简单来说awk就是把文件逐行的读入，以空格为默认分隔符将每行切片，切开的部分再进行各种分析处理。

使用方法：awk’{pattern + action}’{filenames}

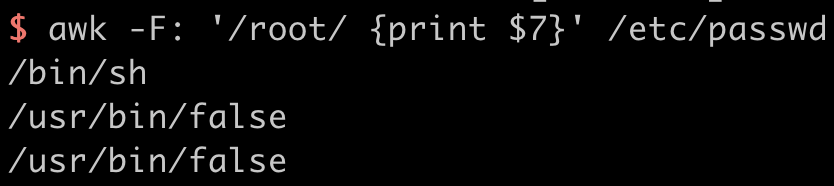
pattern表示表示AWK在数据中查找的内容，而action是在匹配内容时所执行的一系列命令。花括号{}不需要在程序中始终出现，但它们用于根据特定的模式对一系列指令进行分组。 pattern就是要表示的正则表达式，用斜杠括起来。比如awk ‘/root/’ /etc/passwd

awk语言的最基本功能是在文件或者字符串中基于指定规则浏览和抽取信息，awk抽取信息后，才能进行其他文本操作。完整的awk脚本通常用来格式化文本文件中的信息。通常，awk是以文件的一行为处理单位。awk每接收文件的一行，然后执行相应的命令，来处理文本。

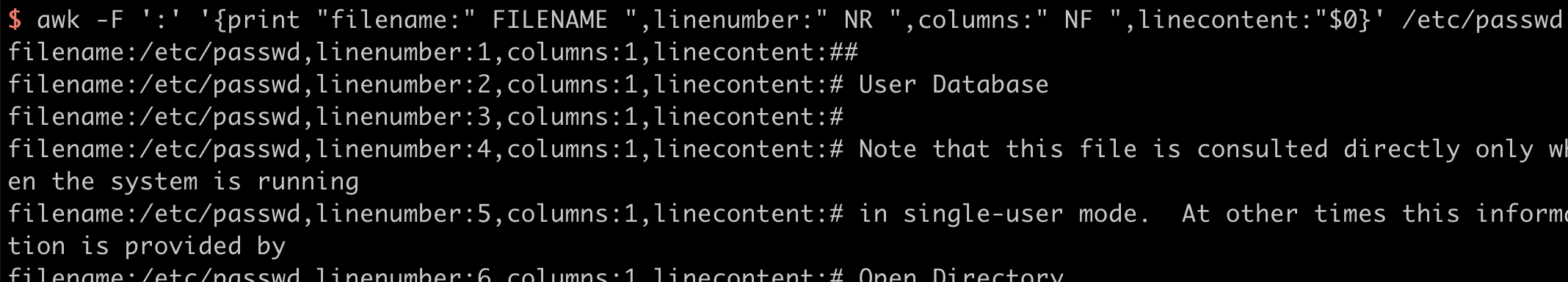
1、搜索/etc/passwd有root关键字的所有行：



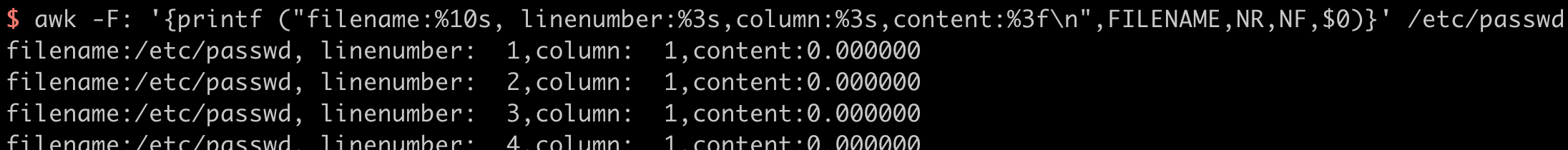
2、搜索/etc/passwd有root关键字的所有行，并显示对应的shell：



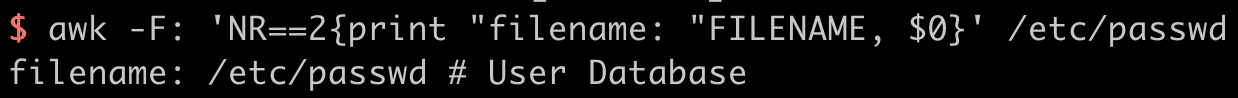
3、统计/etc/passwd:文件名，每行的行号，每行的列数，对应的完整行内容：



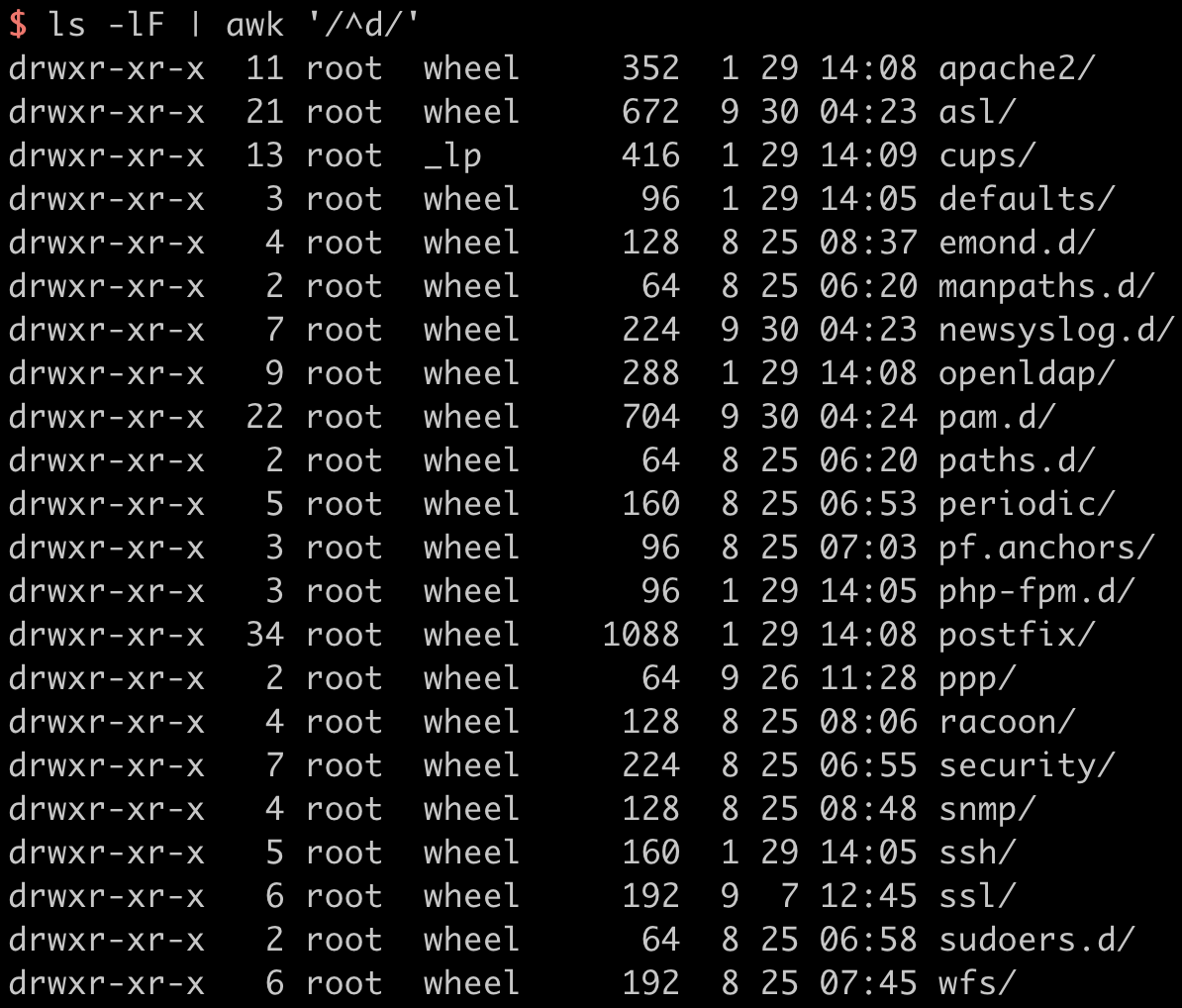
4、使用printf替代print，可以让代码更加简洁，易读：



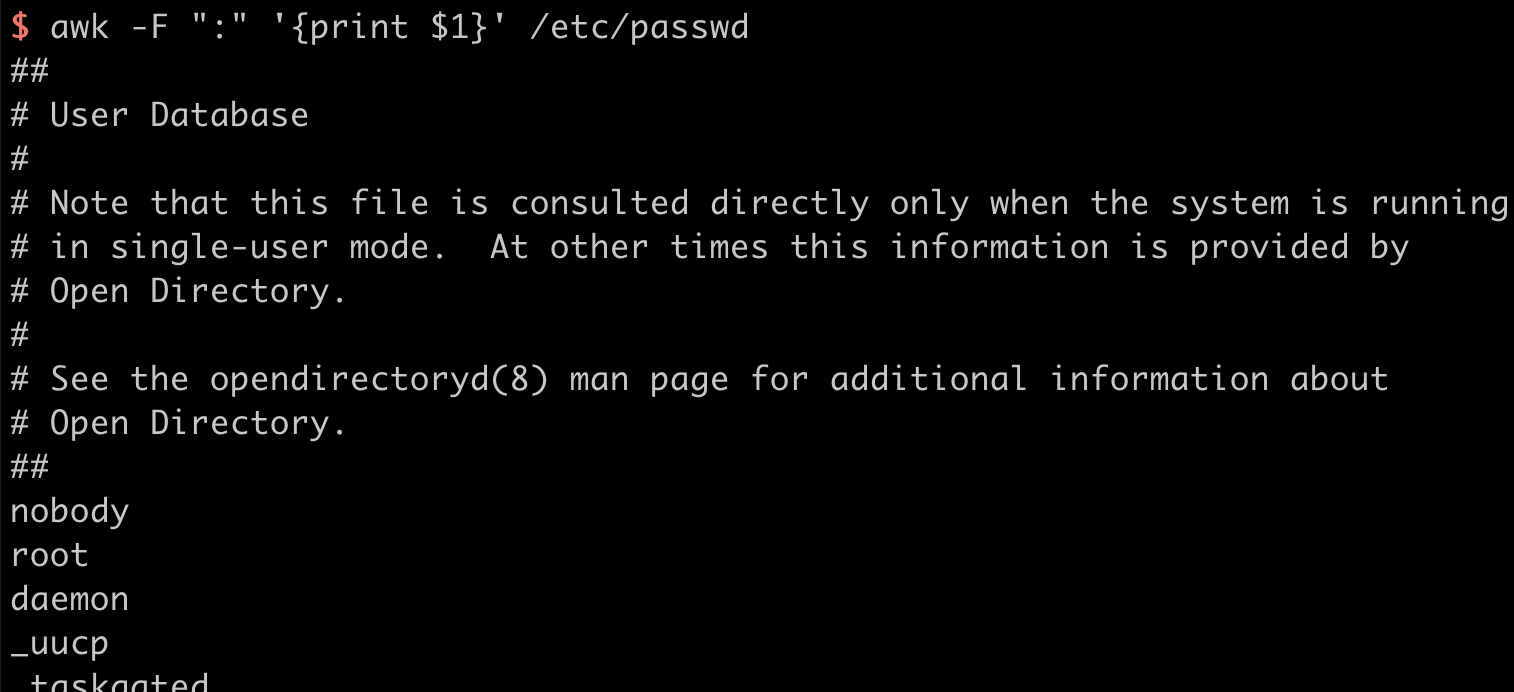
5、打印/etc/passwd/的第二行信息：



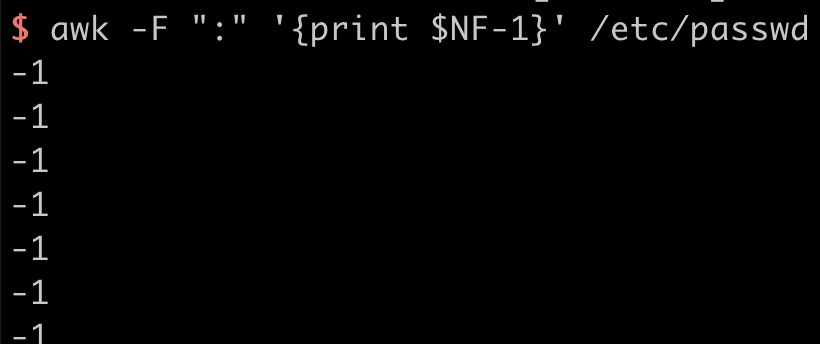
6、awk的过滤使用方法：



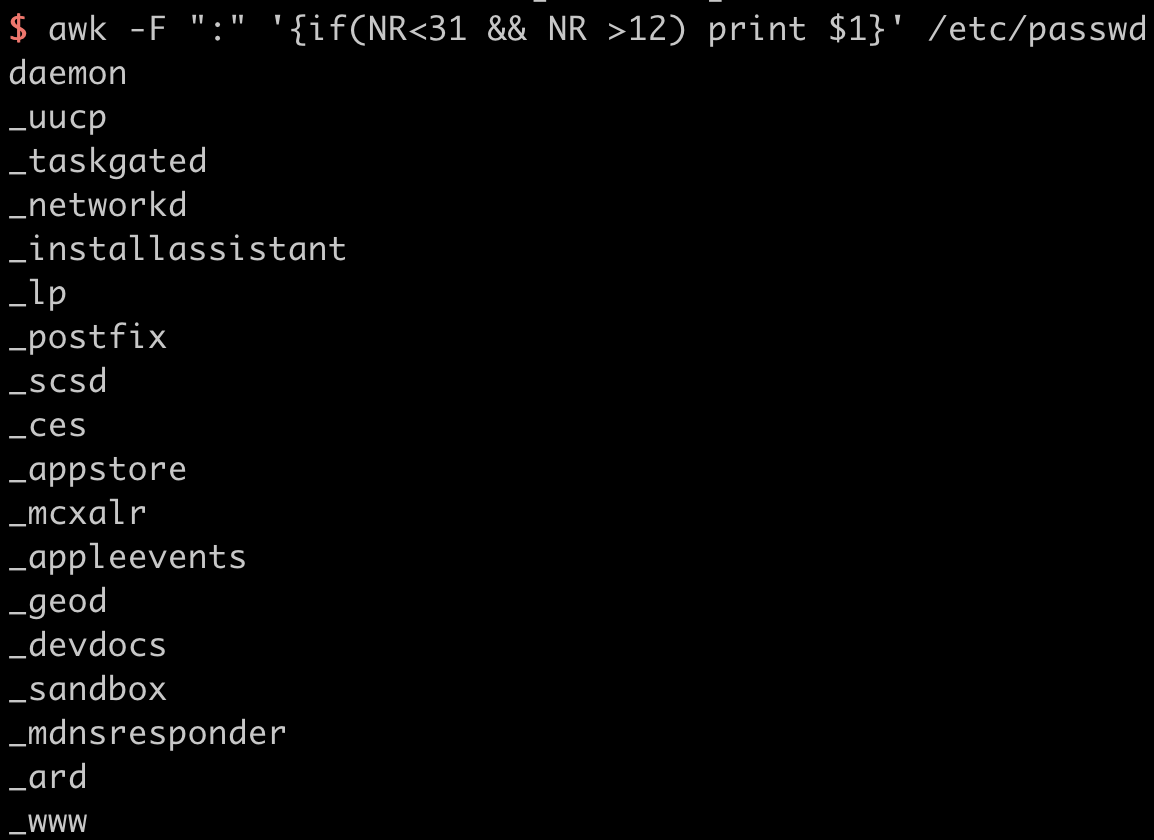
7、指定特定的分隔符，查询第一列：



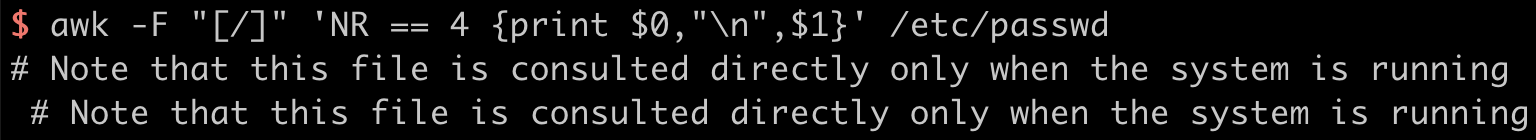
8、指定特定的分隔符，查询倒数第二列：



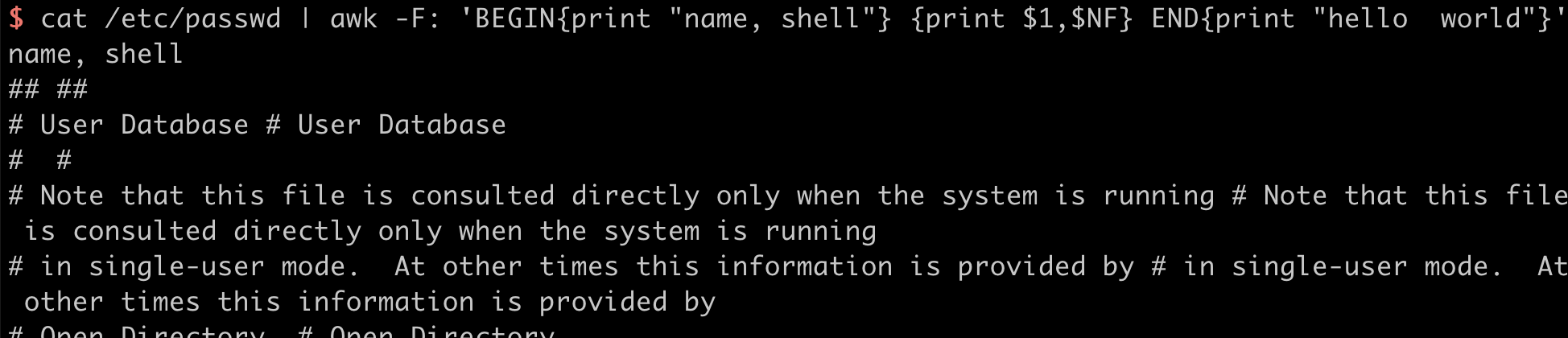
9、获取第12到31行的第一列的信息：



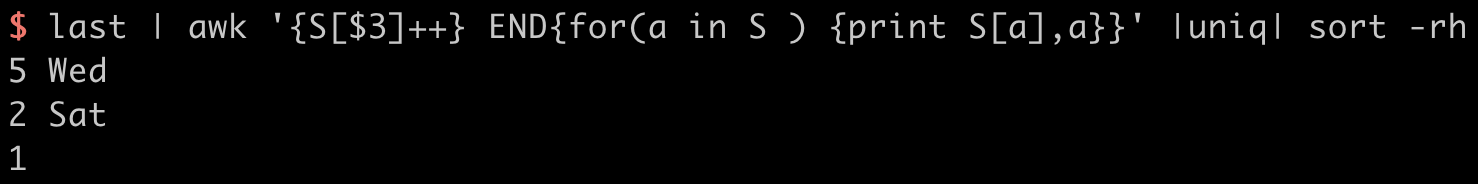
10、多分隔符的使用：



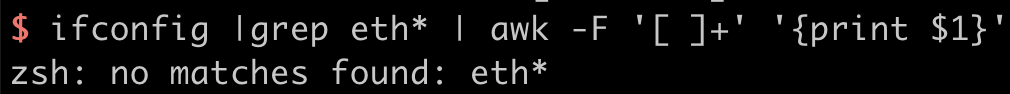
11、添加了BEGIN和END：



12、查看最近登录最多的IP信息：



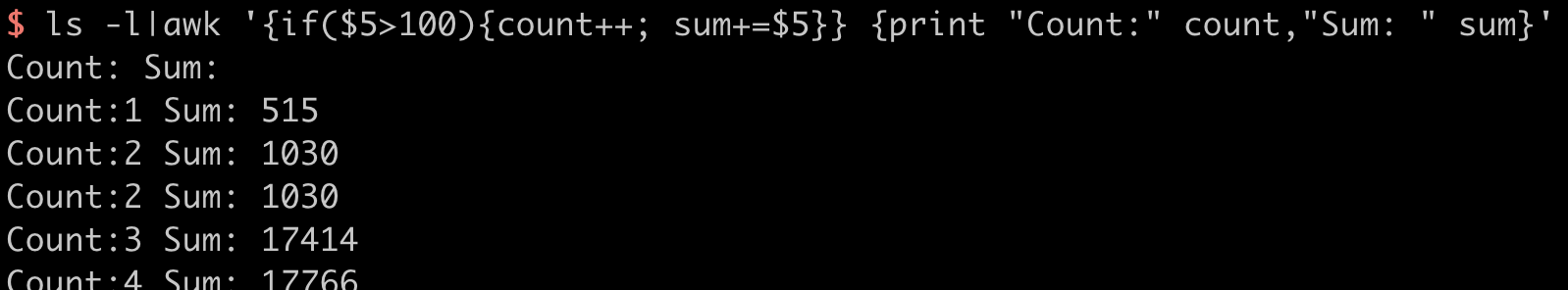
13、利用正则过滤过个空格：



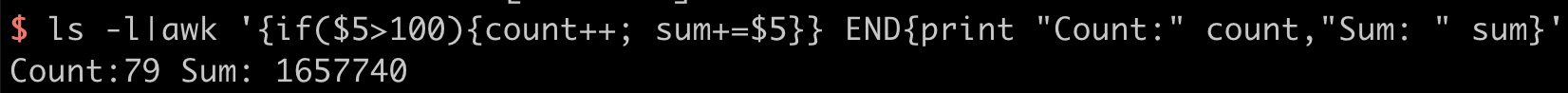
14、awk编程：

除了awk的内置变量，awk还可以自定义变量，awk中的循环语句同样借鉴于C语言，支持while、do/while、for、break、continue，这些关键字的语义和C语言中的语义完全相同。

统计某个文件夹下的大于100k文件的数量和综合。

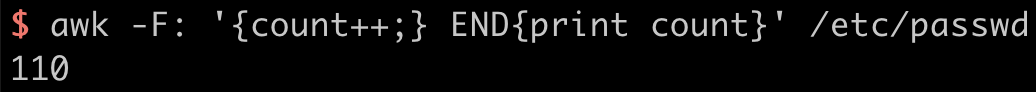


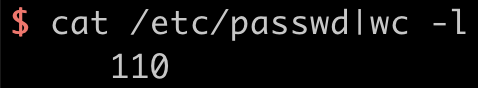
awk会轮询统计，所以会显示整个过程。

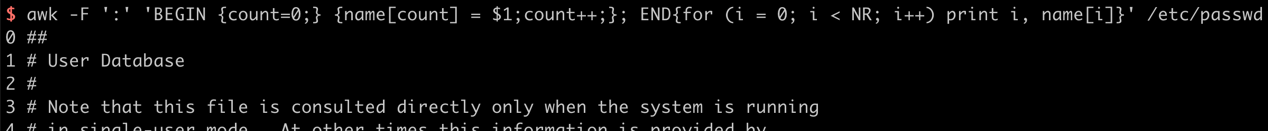


添加END后只显示最后的结果。

统计显示/etc/passwd的账户：







#### top

Linux中的top命令就像是Windows中的任务管理器。它会以列表的形式展示出系统的当前状态以及进程信息，并且定时刷新，同时也支持一些交互性的操作。



第一行：概况

第二行：进程计数（Task）

第三行：CPU使用率（%Cpu(s)）

第四、五行：物理内存和交换空间（Mem/Swap）

以下所有行：进程信息信息

top命令的界面中，可以输入一些指令实现交互性的操作，下面列出一些比较常用的交互性操作：

CTRL+L：刷新整个屏幕，重新开始显示。

h：显示帮助。

q：退出top程序。

空格：立即刷新信息。

k：杀掉进程。输入k之后，会提示用户输入PID及要发送哪种信号。

r：重新设置进程优先级，即renice。输入r之后，会提示用户输入PID及新的nice值。

s：改变刷新周期。输入s之后，会提示用户输入新的刷新周期，单位为秒。

n：改变进程列表中的显示数量。

f：添加或删除进程列表中的列。输入f之后会显示字母与列的映射表，再输入对应字母就可以开关相应的列。

c：在COMMAND列中切换显示命令名和完整的命令行。

u：指定在进程列表中只显示对应用户的进程。

l：切换显示界面中第一行信息（时间和平均负载）。

t：切换显示界面中第二、三行信息（进程和CPU统计）。

m：切换显示界面中第四、五行信息（内存和交换空间）。

1：切换展开CPU统计信息。展开后，会分别显示CPU每个逻辑核心的占用。

N：按PID对进程排序。

M：按%MEM对进程排序。

P：按%CPU对进程排序。

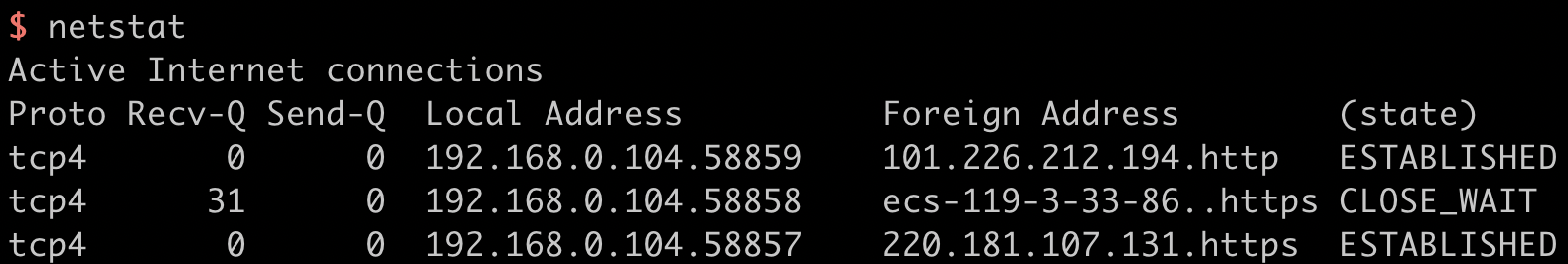
T：按TIME+对进程排序。

H：切换在进程列表中显示所有线程信息。

其中有些操作可以直接跟在top命令后来使用，比如top -c、top -H等。另外还有一个top -p，用来指定PID显示。由于它支持逗号分隔的多个PID，因此用法很灵活。

#### netstat

监控CP/IP网络的工具，它可以显示路由表、实际的网络连接以及每一个网络接口设备的状态信息。Netstat用于显示与IP、TCP、UPD和ICMP协议相关的统计数据，一般用于检验本机各端口的网络连接情况。



输出结果可以分为两个部分：

* Active Internet connections有源TCP连接，“Recv-Q”和“Send-Q”指接收队列和发送队列。一般都为0，如果不是则表示软件包正在队列中堆积。这种情况只能在非常少的情况下见到。
* Active UNIX domain sockets有源Unix域套接口（和网络套接字一样，但是只能用于本机通信，性能可以提高一倍）。

1、列出所有端口 #netstat -a

2、列出所有 tcp 端口 #netstat -at

3、列出所有 udp 端口 #netstat -au

4、只显示监听端口 #netstat -l

5、只列出所有监听 tcp 端口 #netstat -lt

6、只列出所有监听 udp 端口 #netstat -lu

7、列出所有监听 UNIX 端口 #netstat -lx

8、显示所有端口的统计信息 #netstat -s

9、显示 TCP 或 UDP 端口的统计信息 #netstat -st 或 -su

10、 输出中显示 PID 和进程名称 #netstat -p

11、netstat 输出中不显示主机，端口和用户名 (host, port or user)

当你不想让主机，端口和用户名显示，使用 netstat -n。将会使用数字代替那些名称。

同样可以加速输出，因为不用进行比对查询。

#netstat -an

如果只是不想让这三个名称中的一个被显示，使用以下命令

# netsat -a --numeric-ports

# netsat -a --numeric-hosts

# netsat -a --numeric-users

12、持续输出 netstat 信息  #netstat -c

13、找出程序运行的端口 #netstat  -ap | grep ':80'

14、查看连接某服务端口最多的的IP地址（前20个）

#netstat -nat | grep "10.1.62.23:443" |awk '{print $5}'|awk -F: '{print $1}'|sort|uniq -c|sort -nr|head -20

15、TCP各种状态列表

#netstat -nat |awk '{print $6}'

统计数量

#netstat -nat |awk '{print $6}'|sort|uniq -c

排序

#netstat -nat |awk '{print $6}'|sort|uniq -c|sort -rn

#netstat -n | awk '/^tcp/ {++S[$NF]} END {for(a in S) print a, S[a]}'

16、直接统计tcp数量监听的数量

#netstat -ant | wc -l

#### grep（Global Regular Expression Print）

文本搜索工具，使用正则表达式搜索文本，并把匹配的行打印出来。使用权限是所有用户。grep家族包括grep、egrep和fgrep。egrep和fgrep的命令只跟grep有很小不同。egrep是grep的扩展，支持更多的re元字符， fgrep就是fixed grep或fast grep，它们把所有的字母都看作单词，也就是说，正则表达式中的元字符表示回其自身的字面意义，不再特殊。linux使用GNU版本的grep。它功能更强，可以通过-G、-E、-F命令行选项来使用egrep和fgrep的功能。

#### less

less工具对文件或其它输出进行分页显示的工具，是linux正统查看文件内容的工具。

默认查看文件内容：

less /etc/passwd

ps查看进程信息并通过less分页显示，同时显示行号：

ps -ef | less -N

#### tail

依照要求将指定的文件的最后部分输出到标准设备，通常是终端，通俗讲来，就是把某个档案文件的最后几行显示到终端上，假设该档案有更新，tail会自己主动刷新，确保你看到最新的档案内容。

显示passwd文件的尾部内容（默认10行，相当于添加参数-n 10），刷新显示在屏幕上，退出，按下ctrl+c：

tail -f /etc/passwd

### 4内存分页管理与Swap(很少考)

### 5、任务队列与CPU Load

#### 6、扩展

#### 内存屏障

#### 指令乱序

#### 分支预测

#### CPU亲和性（affinity）

#### Netfilter与iptables