Linux系统管理

**进程管理**

什么是程序？

代替人管理操作系统，完成特定任务，解决特定问题的一段代码集合。

什么是进程？

进程（Process）是计算机中的程序关于某数据集合上的一次运行活动。

什么是线程？

线程是进程中的一个实体，被称为轻量级进程，是程序执行流的最小单元。

进程的作用？

1. 判断服务器的健康状态：我们可以通过观察服务器的CPU 内存等占用率来提前发现问题，及时解决，避免发生宕机或者服务卡死！
2. 查看系统内的进程是否合法：我们要经常过滤进程中查看是否有非法进程在运行，避免信息的泄露或损坏
   * 进程的查看
     1. ps 命令

静态查看系统中的进程，查询的是命令执行一瞬间的进程的状态。

aux 以BSD操作系统的格式输出

-le 以Linux标准格式输出

选项详解：

a 显示一个终端的所有进程

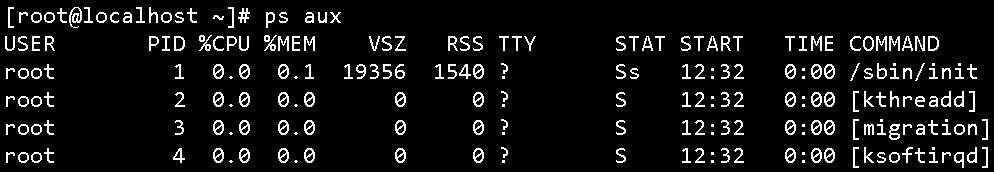
u 显示进程的归属用户及内存的使用情况

x 显示没有控制终端的进程

l 长格式显示，更加详细的信息

e 显示所有进程 与a的作用一致

结果分析：



USER： 该进程的运行用户，或者理解为产生进程的用户

PID： 进程的PID号

%CPU： 进程占用的cpu的百分比

%MEM： 进程占用物理内存的百分比

VSZ： 占用的虚拟内存的大小（KB）

RSS： 占用物理内存的大小（KB）

tty： 登录终端（Alt+F1-F6）

STAT：

D 无法中断的休眠状态（通常 IO 的进程）

R 正在运行的进程

S 处于休眠状态

T 停止或被追踪

W 进入内存交换（从内核2.6开始无效）

X 死掉的进程（基本很少见 ）

Z 僵尸进程

< 优先级高的进程

N 优先级较低的进程

L 被锁进内存的进程

s 进程的领导者（在它之下有子进程）

l 多进程的

+ 位于后台的进程组

START： 进程启动时间

TIME： 进程运行的时间（总时间），非系统时间

COMMAND： 产生此进程的命令

* + 1. ps -le 结果分析

F： 进程标志，说明进程有那些权限

1： 进程可以复制

4： 进程使用超级用户权限

S： 进程状态

PPID： 进程的父进程ID号

C： cpu占比 %

PRI： 进程优先级，数字越小优先级越高，越快被调用执行

NI： 进程优先级，数字越小越先被调用

ADDR： 进程存在内存的位置

SZ： 进程占用的内存大小

WCHAN： 进程是否运行（-代表正在运行）

* + 1. top 命令

选项：

-d 秒数 指定刷新频率

-b 使用批处理模式，和-n结合将top的结果重定向到文本

-n 次数 指定top 次数，和-b结合

-p 指定查看某的PID的进程

-s 使top在安全模式下运行，避免在交互模式下出现错误

-u 用户名 只查看（监听）某一个用户的进程

在交互模式下使用的命令：

？或h 帮助信息

P 按照CPU使用率排序

M 按照内存使用率排序

N 按照PID号排序

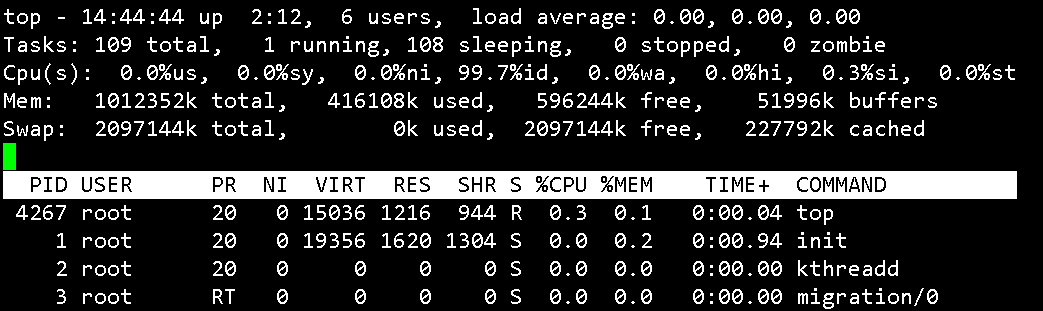
T 按照cpu运算时间排序

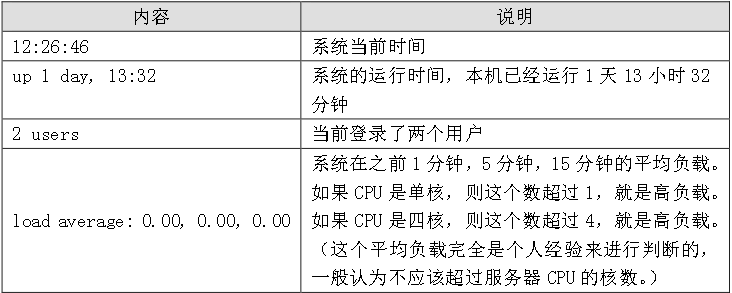
k 按照PID号给予某一个进程信发送指令，可用于杀死进程

r 按照PID给某个进程重设优先级（Nice）值

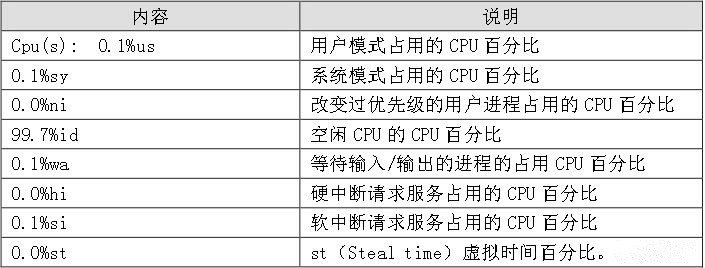
q 退出交互模式

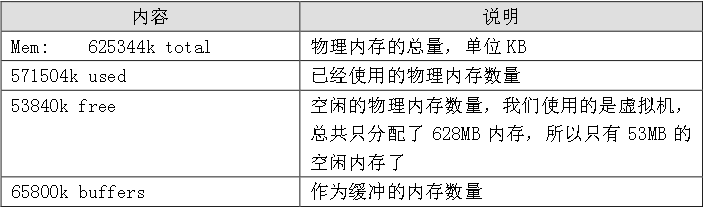
top 结果分析

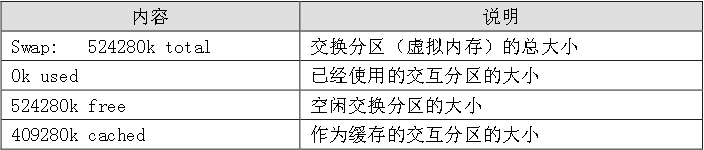


第一行信息：

第二行信息：

第三行信息：

第四行信息：

第五行信息：

剩下的选项基本和我们的PS是相同的。

拓展：pstree 进程树

-p 显示PID

-u 显示运行用户

* + 进程处理命令：

1. kill 进程号

kill -l #查看kill 所支持的信号

常用的信号：

HUP（1）： 复位，重新加载配置

INT （2）： 中断（同 Ctrl + C）

TERM（15）： 终止（正常退出）

KILL（9）： 强制终止

CONT（18）：继续（与STOP相反）

STOP（19）： 暂停（同 Ctrl + Z）

ps aux | grep “vi” | grep -v“grep”

2. killall 进程名

killall -i #交互式，询问是否杀死

killall -I（大） #忽略大小写

3. pkill 终端号

pkill -9 -t 终端号 #强制杀死某登录终端

* + 进程优先级

ps –le 可以查看到进程的优先级

PRI代表的是Priority，NI代表nice 两个都是优先级，但是PRI不能修改，因为这是内核内定义的，无法人为修改，但是我们能修改NI 。

公式：

PRI（最终值）=PRI（原值）+NI

我们可以通过这样的方式改变优先级，但是NI也有限制。

NI值总范围：-20 到 19

普通用户的NI值 : 0 到 19，普通用户只能调高NI值，不能降低；root可以随意调整，包含其他用户的进程

nice 命令（只能在启动或重启时）

nice –n -5 service httpd start

ps –le | grep “httpd” | grep –v “grep”

renice 命令（修改启动状态进程的NI值）

renice -10 PID

**工作管理**

工作类型：

前台：当前终端正在进行标准输出到显示器上的工作

后台：不用将命令执行过程标准输出到显示器的工作

当前终端的工作只能当前终端管理，其它终端不能管理。

放入后台的工作必须是能持续运行的，否则程序会被暂停。

如何将工作放到后台？

1. 命令 & #将工作到到后台，并持续执行

2. ctrl + z #将工作放到后台，并暂停执行

后台工作的管理：

1. 查看： jobs

-l 显示工作进程PID

2. 将后台暂停的工作恢复到前台继续执行

fg %工作号 # %可省略，主要是却别PID的

3. 将后台暂停的工作恢复到后台继续执行

bg %工作号

4. 让后台的工作脱离终端进行运行

a. /etc/rc.local 文件

b. 定时计划任务（下边）

**系统资源查看**

1. vmstat #监控系统资源

Procs（进程）：

r: 运行队列中进程数量

b: 阻塞的进程

Memory（内存）：

swpd: 虚拟内存已使用的大小

free: 空闲的物理内存的大小

buff: 用作缓冲的内存大小（系统）

cache: 用作缓存的内存大小（被调用资源）

Swap：

si: 每秒从磁盘读入虚拟 内存的大小

so: 每秒虚拟内存写入磁盘的大小

注：如果这个值大于0，表示物理内存不够用

IO: （现在的Linux版本块的大小为1024bytes）

bi: 每秒读取的块数

bo: 每秒写入的块数

系统：

in: 每秒CPU中断数，包括时钟中断。

cs: 每秒上下文切换数（资源调用的次数，如：函数）。

CPU（以百分比表示）：

us: 用户进程执行消耗cpu时间(user time)

sy: 系统进程消耗cpu时间(system time)

id: 空闲时间(包括IO等待时间)

wa: 等待IO时间

st: 虚拟机偷取时间（不为0则代表cpu被其他虚拟机所调用）

2. dmesg

查看开始是内核的检测信息（/var/log/dmesg）

3. free 查看内存使用情况

-b 以字节为单位显示

-k 以KB为单位（默认）

-m 以MB为单位

-g 以GB为单位

显示信息解释：

第一行：

total: 总计物理内存的大小

used: 已使用内存大小

free: 空闲内存大小

shared: 共享内存，一般系统不会用到

Buffers/cached: 未使用的缓存大小

第二行：(-/+ buffers/cached)

(-buffers/cache)used内存数：第一部分Mem行中的 used – buffers – cached

(+buffers/cache)free内存数：第一部分Mem行中的 free + buffers + cached

注：可见-buffers/cache反映的是被程序实实在在吃掉的内存，而+buffers/cache反映的是空闲的内存总数。

第三行：swap的前面有！

4. 查看cpu信息

#逻辑CPU个数

cat /proc/cpuinfo | grep "processor" | wc -l

#物理CPU个数：

cat /proc/cpuinfo | grep "physical id" | sort -u | wc -l

#每个物理CPU中Core的个数：

cat /proc/cpuinfo | grep "cpu cores" | uniq | awk -F: '{print $2}'

#查看core id的数量,即为所有物理CPU上的core的个数

cat /proc/cpuinfo | grep "core id" | uniq | wc -l

5. uptime

系统启动时间和平均负载（一般用top）

6. 查看系统与内核相关信息

uname

-a 查看所有相关信息

-r 查看内核版本

-s 查看内核名称

lsb\_release -a

查看当前发行版

7. lsof 列出进程打开或正在使用的文件信息

lsof -p 进程号 #查看某进程运行所调用的资源

lsof -u #查看用户运行的进程调用了哪些文件

**系统定时任务**

1. at：单次计划任务命令

at 命令的执行需要依托atd服务

service atd start

chkconfig atd on

限制用户使用at 命令

/etc/at.allow #白名单

/etc/at.deny #黑名单

白名单的优先级大于黑名单

假如黑白名单都不存在，只有root能使用at命令

at [选项] 时间

-m #执行结束发送通知邮件（执行此命令的用户）

atq #查看计划任务

atrm #删除计划任务

-c 工作号 #查看对应工作号的详细信息

时间格式：

[HH:MM] [yyyy-mm-dd] (now + 2minutes)

写完后用ctrl + d 结束并保存任务

2. crontab 循环计划任务命令

按照预先设置好的时间周期，定期执行特殊动作（命令、脚本）

格式：

编辑：crontab -e #添加循环计划任务

查看：crontab -l #查看已存在的计划任务

删除：crontab -r #删除已存在的计划任务

时间格式：



\* 表示该范围内的任意时间

, 表示间隔的多个不连续时间点

- 表示一个连续的时间范围

/ 指定间隔的时间频率

例子：

0 17 \* \* 1-5 #周一到周五每天17:00

30 8 \* \* 1,3,5 #每周一、三、五的8点30分

0 8-18/2 \* \* \* #8点到18点之间每隔2小时

0 \* \*/3 \* \* #每隔3天