GNU/Linux
系统资源管理命令

系统资源管理命令 -iostat

命令 :iostat

功能 : 输出 CPU 和磁盘 I/O 相关的统计信息 -iostat

语法格式 :iostat [选项] [延迟] [计数]

示例

1. 查看本地服务器的 CPU 与硬盘的信息
   #iostat

系统资源管理命令 -iostat

输出结果说明：

第 1 行：

kernel 版本（完整的主机名称）

报告生成日期

系统架构 (CPU 数）

系统资源管理命令 -iostat

输出结果说明：

第 3 行：

CPU 平均值	行名称	说明
avg-cpu	%user	在用户运行进程所占用的 CPU 百分比
	%nice	进程优先级操作所占用的 CPU 百分比
	%sys	系统级别 (kernel) 运行所使用的 CPU 百分比
	%iowait	CPU 等待硬件 I/O 时所占用的 CPU 百分比
	%idle	CPU 空闲时间的百分比

系统资源管理命令 -iostat

输出结果说明：

第 6 行：

硬盘设备	列名称	说明
Device	tps	每秒钟传输的 IO 请求的数量
	Blk_read/s	块设备每秒钟读取的数量
	Blk_wrtn/s	块设备每秒钟写入的数量
	Blk_read	块设备读出的总数
	blk_wrtn	块设备写入的总数

系统资源管理命令 -iostat

iostat 参数

-c: 仅显示 cpu 信息

-d: 仅显示磁盘信息

-k: 以 k 为单位显示磁盘每秒请求的块数

-t: 显示报告生成时间

系统资源管理命令 -iostat

iostat 参数

-p device | all : 显示指定或所有的块设备信息

-x: 输出扩展信息 [与 -p 参数冲突]

-N: 显示设备映射名

-V: 显示 iostat 版本信息

系统资源管理命令 -iostat

示例：

2. 每 2 秒，显示一次设备统计信息
   #iostat -d 2

3. 每 2 秒显示一次设备统计信息，共计 6 次
   #iostat -d 2 6

4. 每 2 秒以 K 为单位显示一次设备统计信息，且显示 LVM 映射名称，共计 10 次
   #iostat -dNk 2 10

系统资源管理命令 -vmstat

命令 :vmstat

功能 : 监控 CPU 、内存、虚拟内存交换、 IO 读写等各种情况的使用 .

语法格式 :

vmstat [选项] [延迟] [计数]

示例 1:

(1) 显示当前系统的各项信息

#vmstat

系统资源管理命令 -vmstat

输出结果

进程	显示	说明
proc	r	表示运行队列 (即 : 多少进程真正的分配到 CPU)
	b	阻塞队列 (等待资源分配的进程数)

内存	显示	说明
memory	swap	当前 swap 使用 k 数的情况
	free	当前物理内存空闲的 k 数
	buff	内存使用的 buff 总数 , 一般为块设备操作 , 文档权限记录等
	cache	内存使用的 cache 总数 , 一般打开文件 , 运行程序等使用

系统资源管理命令 -vmstat

输出结果

虚拟内存	显示	说明
swap	si	每秒钟从磁盘读入到 swap 的大小 , 不可长期 >0
	so	每秒钟从 swap 写入到磁盘的大小 , 不可长期 >0

块设备 IO	显示	说明
IO	bi	块设备每秒接收到的块数
	bo	块设备每秒发送的块数

系统情况	显示	说明
System	in	系统的每秒中断数总计 (含时钟中断)
	cs	每秒上下文切换的次数 (系统调用 , 环境变化等)

系统资源管理命令 -vmstat

输出结果

cpu 情况	显示	说明
cpu	us	用户 (及优先级) 占用 CPU 时间
	sy	系统 (kernel 级) 占用 CPU 时间
	id	闲置 CPU 时间
	wa	io 等待 CPU 时间
	st	一个虚拟机占用的 CPU 时间 (如 xen/kvm)

系统资源管理命令 -vmstat

示例：

1. 每 2 秒采样 1 次，总计 10 次

```
#vmstat 2 10
```

2. 每 5 秒检测一次，直至手工停止

```
#vmstat 5
```

系统资源管理命令 -pkill

命令 :pkill

功能 : 控制同名程序的所有进程

语法格式 :

pkill [选项] [pattern]

系统资源管理命令 -pkill

参数：
-G gidlist: 仅匹配真实组 ID 在给定列表中的进程。每一个组 ID 可以使用组名称或者数字的组 ID 指定。

-P ppidlist: 仅匹配给定列表中 PPID 的进程。

-t termlist: 仅匹配与给定列表中终端关联的进程。每一个终端指定为在 /dev 中终端设备路径名称的后缀。例如 term/a 或者 pts/0 。

系统资源管理命令 -pkill

参数：
-U uidlist: 仅匹配真实的用户 ID 在给定列表中的进程。

-u euidlist: 仅匹配有效用户 ID 在给定列表中的进程

-signal: 指定发往每一个匹配进程的信号。如果没有指定， SIGTERM 是默认的信号。

-x: 仅认为进程其参数字符串或者执行文件名称正确匹配规定模式是匹配的进程。

系统资源管理命令 -pkill

示例：

1. 将某个终端的用户踢出

```
#pkill   -kill  -t pts/2
```

2. 按用户名踢出用户

```
#pkill -kill -U  arisa
```

3. 强制使 arisa 账户登出

```
#pkill -9 -u arisa
```

系统资源管理命令 -pkill

值得注意，当利用 pkill 对 PPID 进程操作时，实际上仅针对此 PPID 的子进程操作，而不会对指定的 PPID 进行操作。如

示例：

1)snow 账户执行进程

$tty

/dev/tty2

$id -un

snow

$sleep 100000s

系统资源管理命令 -pkill

2)root 操作

#tty

/dev/tty1

#pstree -p snow

```
bash(8391)─┬─sleep(8425)
           ├─sleep(8426)
           └─sleep(8427)
```

#pkill -P 8391

#pgrep -l -u snow

bash(8391)

系统资源管理命令 -pkill

2)root 操作
pkill -SIGKILL -P 8391

# pgrep -l -u snow
bash(8391)

系统资源管理命令 -pgrep

命令 :pgrep
功能 : 程序检查在系统的中活动进程，报告进程属性匹配命令行上指定条件的进程的 ID 。

语法格式 :pgrep [选项] [pattern]

参数与 pkill 一致
-l 长格式输出 , 仅 pgrep 有效

系统资源管理命令 -pgrep

示例：

1. 获得以 root 账户执行的 sshd 的 PID

```
#pgrep -x -u root sshd
```

2. 显示指定账户 snow 所执行的 PID 及相关名称

```
#pgrep -l  -u  snow
```

系统资源管理命令 -pgrep

命令 :pidstat

功能 : 监控被 Linux 内核管理的独立任务 (进程)

说明 :

它输出每个受内核管理的任务的相关信息。 pidstat 命令也可以用来监控特定进程的子进程。间隔参数用于指定每次报告间的时间间隔。它的值为 0(或者没有参数) 说明进程的统计数据的时间是从系统启动开始计算的。

系统资源管理命令 -pidstat

语法格式 :pidstat [选项]

示例
#pidstat

PID - 被监控的任务的进程号

%usr - 当在用户层执行 (应用程序) 时这个任务的 cpu 使用率，和 nice 优先级无关。注意这个字段计算的 cpu 时间不包括在虚拟处理器中花去的时间。

系统资源管理命令 -pidstat

%system – 这个任务在系统层使用时的 cpu 使用率。

%guest – 任务花费在虚拟机上的 cpu 使用率（运行在虚拟处理器）。

%CPU – 任务总的 cpu 使用率。在 SMP 环境（多处理器）中，如果在命令行中输入 -I 参数的话，cpu 使用率会除以你的 cpu 数量。

CPU - 正在运行这个任务的处理器编号。

Command - 这个任务的命令名称。

系统资源管理命令 -pidstat

CPU – 正在运行这个任务的处理器编号。

Command - 这个任务的命令名称。

系统资源管理命令 -pidstat

示例 2

显示 PID2864 的 I/O 信息

#pidstat -d -p 2864

kB_rd/s - 任务从硬盘上的读取速度（ kb ）
kB_wr/s - 任务向硬盘中的写入速度（ kb ）
kB_ccwr/s - 任务写入磁盘被取消的速率（ kb ）

系统资源管理命令 -pidstat

示例 3

显示 PID2864 的内存使用情况的数据。

#pidstat -r -p 2864

inflt/s – 从内存中加载数据时每秒出现的小的错误的数目，这些不要求从磁盘载入内存页面。

majflt/s - 从内存中加载数据时每秒出现的较大错误的数目，这些要求从磁盘载入内存页面。

系统资源管理命令 -pidstat

示例 3

VSZ – 虚拟容量：整个进程的虚拟内存使 (kb)

RSS - 长期内存使用：任务的不可交换物理内存的使用量 (kb)

系统资源管理命令 -pidstat

示例

1. 显示 5 次 page faults 的统计数据结果，每次间隔 2 秒。

```
# pidstat -r 2 5
```

2. 显示所有 mysql 服务器的子进程

```
# pidstat -T CHILD -C mysql
```

3. 统计所有数据并生成一个新的报告

```
# pidstat -urd -h
```

系统资源管理命令 -killall5

命令 :killall5
功能 : 控制系统中的所有进程

语法格式 :killall5 [信号]　进程名

示例 :
关闭所有进程
[root@dgtraing root]# killall5 -9

系统资源监控 -AIDE

命令 :AIDE

全名 :Adevanced Intrusion Detection Environment

中文名称 : 高级入侵检测环境

功能 : 主要用途是检查文档的完整性

系统资源监控 -AIDE

特点：

1. AIDE 能够构造一个指定文档的数据库，

2. AIDE 数据库能够保存文档的各种属性，包括：
   (1) 权限 (permission)
   (2) 索引节点序号 (inode number)
   (3) 所属用户 (user) 、所属用户组 (group)

系统资源监控 -AIDE

特点：

（4）文档大小、最后修改时间 (mtime) 、创建时间 (ctime) 、最后访问时间 (atime)

（5）增加的大小连同连接数。

（6) AIDE 还能够使用下列算法：sha1 、 md5 、 rmd160 、 tiger ，以密文形式建立每个文档的校验码或散列号。

系统资源监控 -AIDE

1. 安装 AIDE

   #yum instsall aide -y

2. AIDE 的配置文件及所在路径

   #vim /etc/aide.conf

3. 对 AIDE 的配置文件进行检测

   #aide -D

系统资源监控 -AIDE

4. AIDE 权限说明

AIDE 的每个检测权限都是由若干的权限组成，其组成的权限有：

p 权限

i i 节点（索引节点）

n 连接数量

u 用户

系统资源监控 -AIDE

4. AIDE 权限说明

g 用户组

s 大小

m 最后一次修改时间

a 最后一次访问时间

c 创建时间

S 检查增加的大小

系统资源监控 -AIDE

4. AIDE 权限说明

l 链接名

b 块计算

l 忽略改变的文件

sha1 sha1 校验

sha256sha256 校验

sha512sha512 校验

系统资源监控 -AIDE

4. AIDE 权限说明

md5 md5 校验

crc32　crc32 校验

rmd160　校验

tiger tiger 校验

haval　haval 校验

系统资源监控 -AIDE

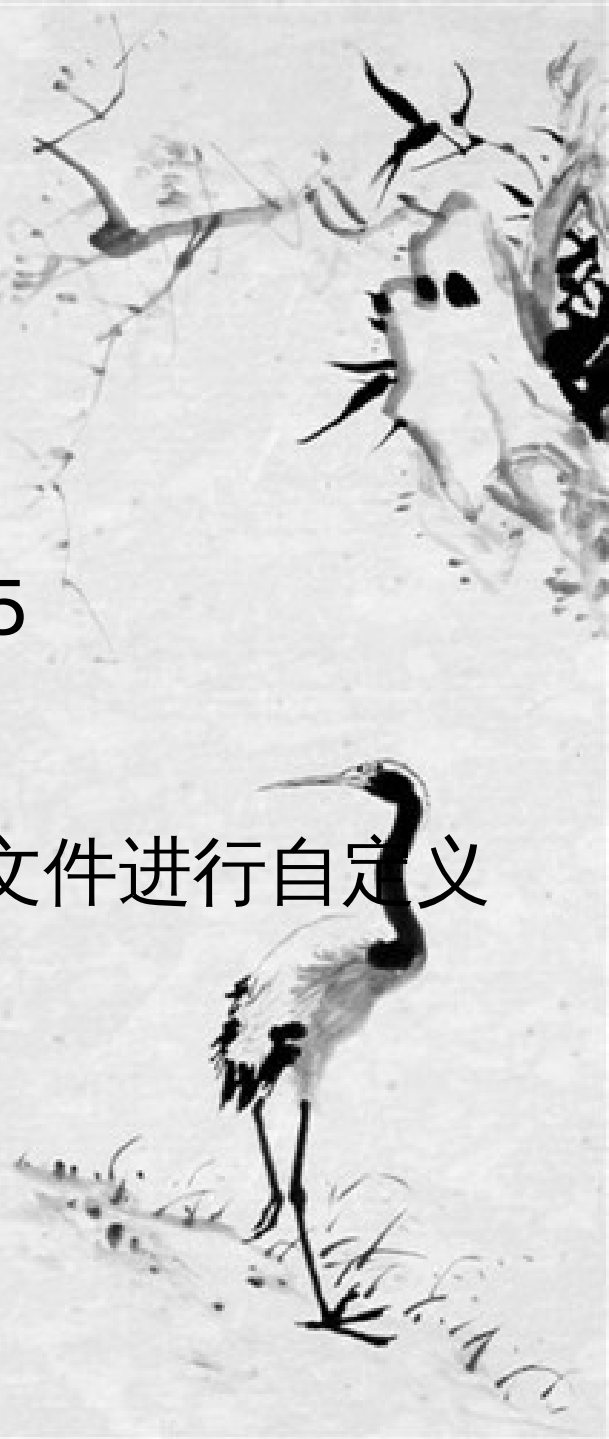
4. AIDE 权限说明

因此 AIDE 的检测权限则为：

R　p+i+n+u+g+s+m+c+md5

L　p+i+n+u+g

亦可以自行定义及对所检测的目录 / 文件进行自定义



系统资源监控 -AIDE

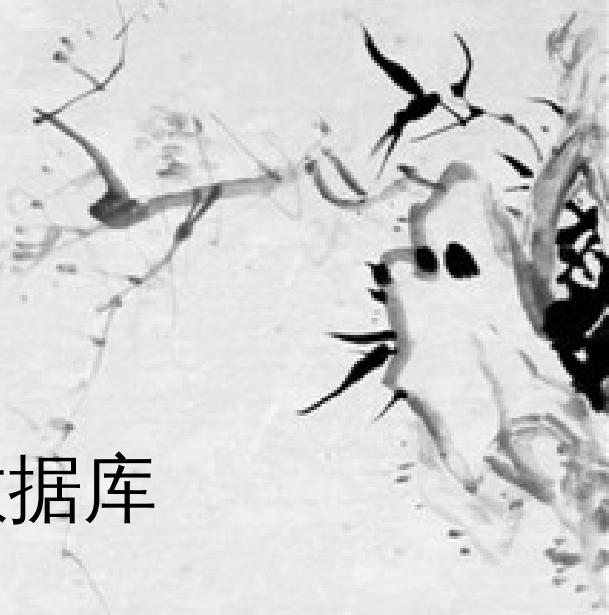
5. 建立数据库

1) 根据 aide.conf 的配置初始化数据库

```
#aide -c /etc/aide.conf -i
```

2) 将新的初始化的数据库进行使用

```
#cd /var/lib/aide
#cp aide.db.new.gz aide.db.gz
```



系统资源监控 -AIDE

6. 检测 / 更新数据库

1) 对系统进行检测

```
#aide  --check
```

或

```
#aide -C
```

2) 如文件修改，对 AIDE 数据库更新

```
#aide  --update
```

或

```
#aide -u
```

系统资源监控 -sar

命令 :sar

全称 :System Activity Reporter

中文名 : 系统活动情况报告

功能 :Linux 系统性能分析工具

系统资源监控 -sar

特点：

可分析 Linux 系统的：

1. 文件的读写情况
2. 系统调用的使用情况
3. 磁盘 I/O
4. CPU 使用率
5. 内存使用
6. 进程活动等

系统资源监控 -sar

语法格式 :

sar [options] [-A] [-o file] -t [n]

参数

-t n1 n2: 指定 n1 的采样间隔时间 n1(秒), 及总共采样次数 n2

-o: 以 2 进制格式记录至指定文件中

-A: 所有采样报告的总和

-u: 采样并输出 CPU 信息

系统资源监控 -sar

参数：

-v: 输出

dentunusd: 目录高速缓存中未使用的缓存条目编号

file-nr: 文件句柄的使用量

inode-nr:i 节点使用量

pty-nr: 伪终端的 pty 数量

-d: 输出每个设备的活动信息

系统资源监控 -sar

参数：

-r: 内存与 swap 的统计信息

-b: 显示 IO 与传送率的统计信息

-a: 文件的读写情况

-c: 进程统计情况，每秒创建的进程数目

-R: 输出内存页的统计信息

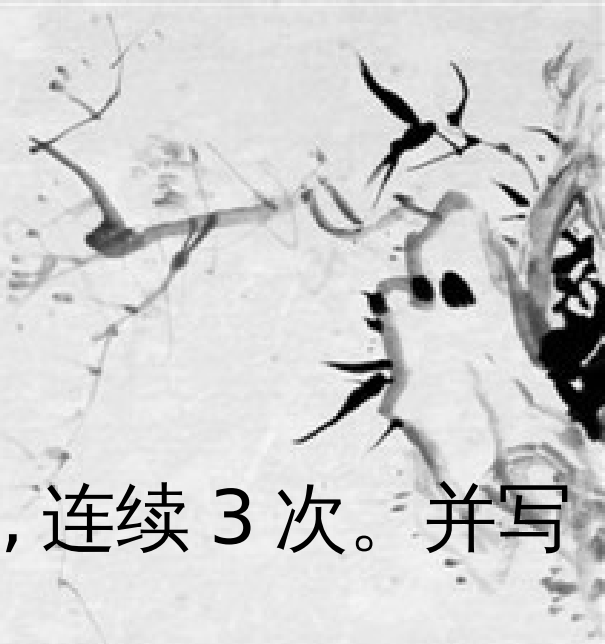
系统资源监控 -sar

参数：

-y: 终端设备活动情况统计信息

-w: 输出系统交换活动的统计信息

系统资源监控 -sar



示例：

1. 对 CPU 使用情况每 10 秒采样 1 次，连续 3 次。并写入文件 test 中 .

```
#sar -u -o test -t 10 3
```

2. 查看 2 进制文件 test

```
#sar -f test
```

系统资源监控 -sar

输出结果说明：

列名	列说明
CPU	所统计的 CPU 数 ,all 为所有
%user	显示 user 占用 CPU 百分比
%nice	显示进程优先级占用 CPU 百分比
%system	显示 kernel 函数 (系统调用) 占用 CPU 百分比
%iowait	显示 IO 操作等待所占用的 CPU 百分比
%steal	显示管理程序为虚拟进程提供服务而等待虚拟 CPU 所占用百分 (xen)
%idel	CPU 闲置百分比

系统资源监控 -sar

输出结果说明：

行名	行说明
报告生成时间	
average	各列项的平均值

系统资源监控 -sar

示例：

3. 内存和 swap 监控

#sar -r

输出结果：

列名	列说说明
kbmemfree	与 free 命令中的 free 值基本一致 , 不包括 buffer 和 cache 的空间
kbmemused	与 free 命令中 used 值一致 , 包括 buffer 和 cache 所使用的值
%memused	物理内存使用的百分比 , 不包含 swap
kbbuffer	等同 free 命令中的 buffer 值
kbcache	等同 free 命令中的 cache 值
kbcommit	确保系统 (防止内存溢出) 所需要的内存数量 (物理内存 +SWAP)
%commit	需要内存数量与总内存 (物理内存 +swap) 的百分比

系统资源监控 -sar

输出结果说明：

行名	行说明
报告生成时间	
averages	各列项的平均值

系统资源监控 -sar

示例：

3. 内存分页监控

#sar -B

输出结果：

列名	列说说明
pgpgin/s	从磁盘或 SWAP 置换到内存每秒的 KB 数
pgpgout/s	从内存置换到磁盘或 SWAP 每秒 KB 数
fault/s	每秒钟系统产生的缺页数 (主缺页 + 次缺页) 之和 (缺页 : 数据未在内存内)
majflt/s	每秒钟产生的主缺页
pgfree/s	每秒被放入空闲队列中的页个数
pgscank/s	每秒被 kswapd 扫描 (置换到 swap) 的页个数
pgscand/s	每秒直接被扫描的页个数

系统资源监控 -sar

示例：

输出结果：

列名	列说说明
pgsteal/s	每秒钟从 cache 中清除出来满足内存需求的页个数
%vmeff	每秒清除的页个数占总扫描页 (pgscank/s+gpscand) 个数的百分比

系统资源监控 -sar

输出结果说明：

行名	行说明
报告生成时间	
average	各列项的平均值

系统资源监控 -sar

示例：

5. IO 与传送速率监控

#sar -b

输出结果：

列名	列说说明
tps/s	每秒钟物理设备 IO 传输总量
rtps/s	每秒钟对物理设备读的数据总量
wtps/s	每秒钟对物理设备写的数据总量

系统资源监控 -sar

示例：

输出结果：

列名	列说说明
bread/s	每秒钟对物理设备读的数据量 (块 /s)
bwrtn/s	每秒钟对物理设备写的数据量 (块 /s)

系统资源监控 -sar

输出结果说明：

行名	行说明
报告生成时间	
average	各列项的平均值

系统资源监控 -sar

示例：

6. 进程队列长度与平均负载状态监控

#sar -q

输出结果：

列名	列说说明
runq-sz	等待运行的进程数 (运行队列的长度)
plist-sz	进程列表中 processes(进程) 和 threads(线程) 的数量
ldavg-1	最后 1 分钟系统的平均负载

系统资源监控 -sar

示例：

输出结果：

列名	列说说明
ldavg-5	最后 5 分钟的系统平均负载数
ldavg-15	最后 15 分钟的系统平均负载数

系统资源监控 -sar

输出结果说明：

行名	行说明
报告生成时间	
average	各列项的平均值

系统资源监控 -sar

示例：

7. 系统交换活动信息监控

#sar -W

输出结果：

列名	列说说明
pswpin/s	每秒系统进入 swap page 的数量
pswpout/s	每秒系统读出 swap page 的数量

系统资源监控 -sar

输出结果说明：

行名	行说明
报告生成时间	
average	各列项的平均值

系统资源监控 -sar

示例：

8. 设备使用情况监控，并显示设备名称而不用设备节点号显示

```
#sar -d  -p
```

输出结果：

列名	列说说明
DEV	设备名称
tps	每秒对物理磁盘 I/O 的次数
rd_sec/s	每秒读取扇区的次数
wr_sec/s	每秒写入扇区的次数

系统资源监控 -sar

示例：

输出结果：

列名	列说说明
avgrq-sz	平均每次设备 I/O 操作的数据大小 (扇区)
avgqu-sz	磁盘请求队列的平均长度
await	从请求磁盘操作到系统完成处理 , 每次请求所消耗的平均时间 (含队列等待时间), 单位为毫秒 (1 秒 =1000 毫秒)
svctm	系统处理每次请求的平均时间 , 不包括请求在队列中所消耗的时间
%util	请求占 CPU 的百分比 , 此数越大 , 越代表设备带宽越饱和

系统资源监控 -sar

输出结果说明：

行名	行说明
报告生成时间	
average	各列项的平均值

系统资源监控 -sar

系统瓶颈查看推荐：

CPU:

```
#sar  -u
#sar  -q
```

内存：

```
#sar  -B
#sar  -W
#sar  -r
```

系统资源监控 -sar

系统瓶颈查看推荐：

设备：

```
#sar  -b
#sar  -u
#sar  -d
```

系统资源监控 -htop

类似 top 的命令：

Htop 是一款非常先进的交互式实时 Linux 进程监测工具。它非常类似 Linux top 命令，但是有一些丰富的功能特性，比如易于使用的界面，可用于管理进程、快捷键、进程的垂直和水平视图以及其他对象。

#yum --enablerepo=epel install htop -y

#htop

系统资源监控 -iotop

iotop ：监测 Linux 磁盘的输入 / 输出

iotop 也非常类似 top 命令和 Htop 程序，但是它有记账功能，可用于监测和显示实时磁盘输入 / 输出及进程。这个工具非常有用，可用于查找具体的进程以及进程的频繁使用的磁盘读取 / 写入操作。

#yum --enablerepo=epel install iotop -y

#iotop

系统资源监控 -iptraf

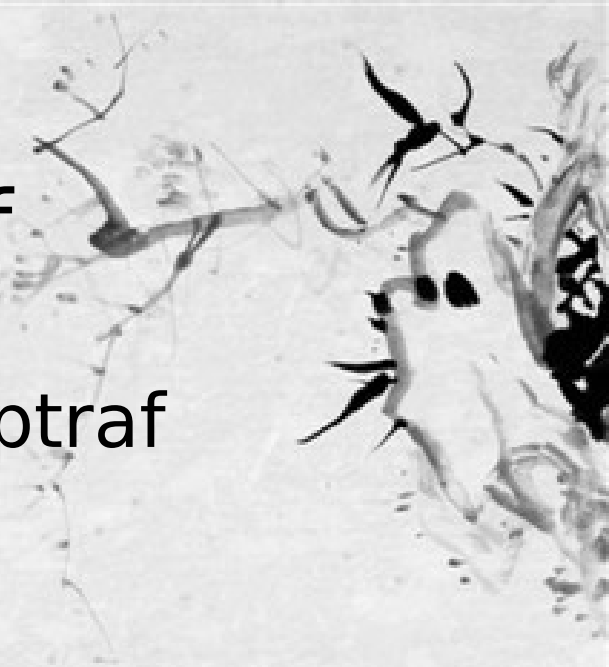
IPTraf ：实时监测 IP 局域网

IPTraf 是一个基于控制台的开源实时网络（IP LAN）监测实用工具，面向 Linux 。它可以收集通过网络传输的众多信息（比如 IP 流量监测器），包括 TCP 标记信息、ICMP 详细信息、 TCP/UDP 流量故障、 TCP 连接数据包以及字节计数。它还可以收集接口方面普通和详细的统计信息，比如 TCP 、 UDP 、 IP 、 ICMP 、非 IP 、 IP 校验和错误以及接口活动等。

系统资源监控 -iptraf

#yum --enablerepo=epel install iptraf

iptraf-ng



系统资源监控 -psacct

监测用户活动

psacct 此工具大有用处，可用于监测系统上每个用户的活动。这两个守护程序都在后台运行，监测系统上每个用户的总体活动，还监测它们在使用什么资源。

这些工具对系统管理员们来说大有用处，可用于跟踪每个用户的活动，比如用户在从事什么操作，他们发出了什么命令，他们使用了多少资源，以及他们在系统上处于活动状态已有多久，等等。

系统资源监控 -psacct

#yum install psacct

#systemctl start psacct
#systemctl enable psacct

系统资源监控 -psacct

1. 现实用户连接时间的统计信息

1) 没有指定参数的 ac 命令会基于来自当前 wtmp 文件的用户登录 / 退出，显示连接时间（小时）的总统计信息
#ac

2) 显示每天的用户统计信息
#ac -d

3) 显示每个用户的时间总数 (小时)
#ac -p

系统资源监控 -psacct

1. 现实用户连接时间的统计信息

4) 显示单个用户时间
#ac snow

5) 显示用户每天的登录时间
#ac -d snow

系统资源监控 -psacct

2. 输出所有账户的活动信息

1) 输出所有用户执行的命令的概要情况
#sa

xxxre 是 " 真实时间 " ，以挂钟分钟为单位。
xxxcp 是系统 / 用户时间（以处理器分钟为单位）之和。
xxxk 是处理器时间平均核心使用，也就是 1k 单位。
第四列是命令名称

系统资源监控 -psacct

2. 输出所有账户的活动信息

2) 输出单个的用户信息
#sa -u

3) 输出进程数量，输出进程总数和处理器分钟总数。如果你看到这些数字不断增大，那么就要查看系统，分析一下出现了什么状况。
#sa -m

系统资源监控 -psacct

2. 输出所有账户的活动信息

4) 输出按百分比排序，显示用户的进程最高百分比。
#sa -c

系统资源监控 -psacct

3. 用户最近执行的命令

1) 搜索和显示之前执行的用户命令信息。你还可以搜索单个用户名称的命令。
#lastcomm snow

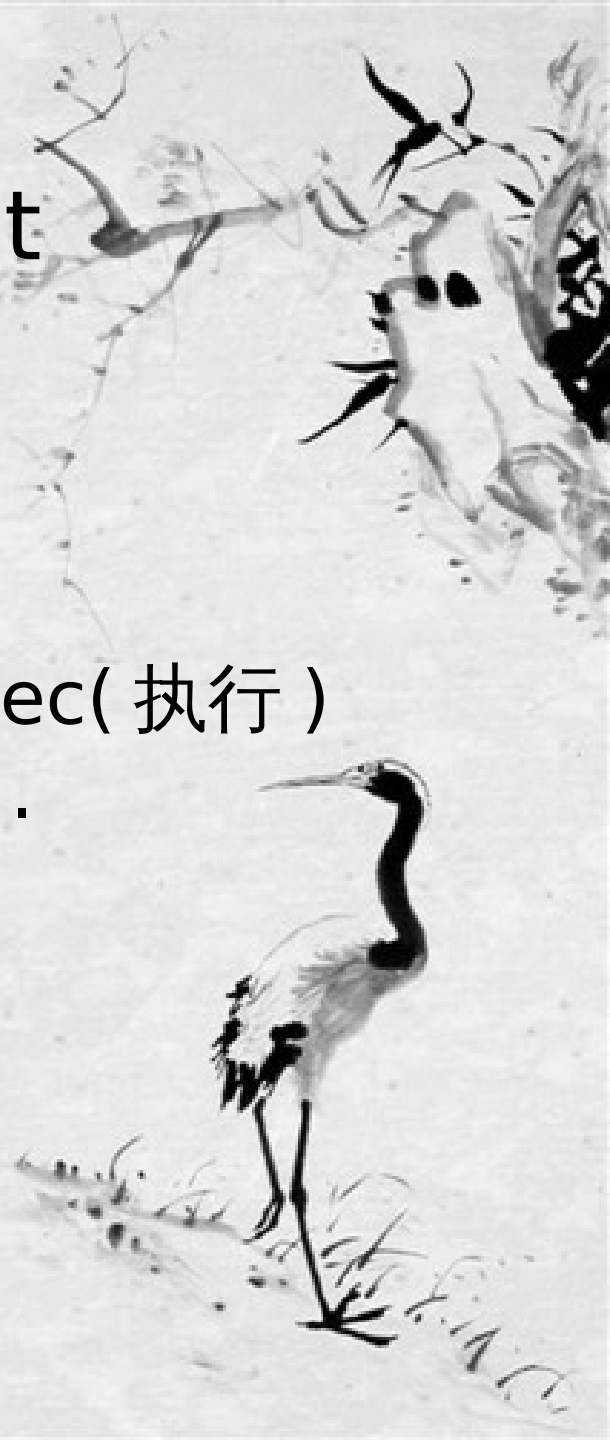
输出分析
第 1 列 : 进程的命令名
第 2 列 :S 和 X 是标志信息，由系统记帐程序管理．每一个标志的含义是 :

系统资源监控 -psacct

3. 用户最近执行的命令

.. S -- 命令由超级用户执行
.. F -- 命令由 fork 产生，但是没有 exec(执行)
.. D -- 命令终止并创建一个 core 文件 .
.. X -- 命令被 SIGTERM 信号终止 .

第 3 列 : 是执行命令的用户名
第 4 列 : 终端名
第 5 列 : 进程退出时间



系统资源监控 -psacct

3. 用户最近执行的命令

2) 搜索命令日志，能够查看每个命令的单独使用情况。
#lastcomm ls

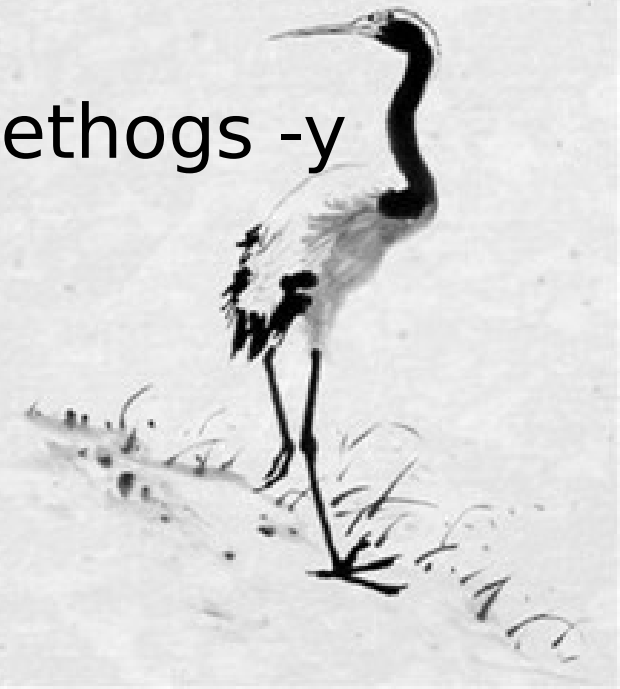
3) 通过终端搜索
#lastcomm pts/0

系统资源监控 -NetHogs

NetHogs 是一款优秀、小巧的开源程序（类似 op 命令），可密切监测系统上每个进程的网络活动。它还密切跟踪每个程序或应用软件所使用的实时网络流量带宽。

#yum --enablerepo=epel install nethogs -y

#nethogs



系统资源监控 -iftop

iftop 是另一款基于终端的免费开源系统监测实用工具，可显示一份经常更新的列表，该列表显示了通过系统上网络接口的网络带宽使用情况。 iftop 通常用于监测网络使用情况，就像 top 通常用于监测处理器使用情况。 iftop 是属于 top 家族的工具，可监测某个所选择的接口，并显示两个主机之间目前的带宽使用情况。

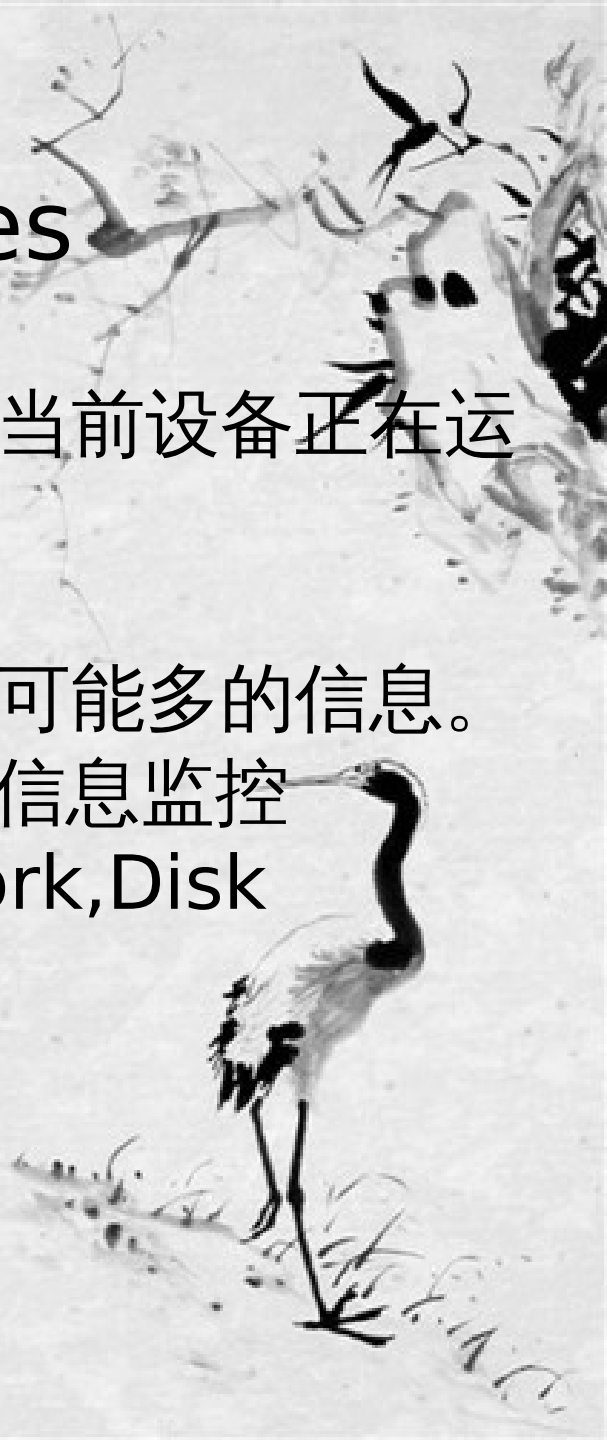
#yum --enablerepo=epel install iftop -y

#iftop
#iftop -i eth0

系统资源监控 -Glances

Glances 用各个分离的表列展示了当前设备正在运行的各种有用的实时数据。

Glances 旨在用最小的空间显示尽可能多的信息。它可以用有限的交互可能性和更深层的信息监控 PerCPU,Load,Memory,Swap,Network,Disk I/O,Mount data 和 Processes



系统资源监控 -Glances

#yum --enablerepo=epel install glances -y

#glances