psutil

**一、psutil模块**

1. **psutil是一个跨平台库（**[**http://code.google.com/p/psutil/**](http://code.google.com/p/psutil/)**），能够轻松实现获取系统运行的进程和系统利用率（包括CPU、内存、磁盘、网络等）信息。**

**它主要应用于系统监控，分析和限制系统资源及进程的管理。它实现了同等命令行工具提供的功能，如ps、top、lsof、netstat、**

**ifconfig、who、df、kill、free、nice、ionice、iostat、iotop、uptime、pidof、tty、taskset、pmap等。目前支持32位和64位的Linux、**

**Windows、OS X、FreeBSD和Sun Solaris等操作系统，**

**2.安装**

wget https://pypi.python.org/packages/source/p/psutil/psutil-2.0.0.tar.gz

tar -xzvf psutil-2.0.0.tar.gz

cd psutil-2.0.0

python setup.py install

或是：pip install psutil

**3.使用**

获取系统性能信息（CPU,内存，磁盘，网络）

3.1CPU相关

查看cpu信息

import Psutil

查看cpu所有信息

>>> psutil.cpu\_times()

scputimes(user=11677.09, nice=57.93, system=148675.58, idle=2167147.79, iowait=260828.48, irq=7876.28, softirq=0.0, steal=3694.59, guest=0.0, guest\_nice=0.0)

显示cpu所有逻辑信息

* psutil.cpu\_times(percpu=True)

[scputimes(user=11684.17, nice=57.93, system=148683.01, idle=2168982.08, iowait=260833.18, irq=7882.35, softirq=0.0, steal=3697.3, guest=0.0, guest\_nice=0.0)]

查看用户的cpu时间比

* psutil.cpu\_times().user 11684.4

查看cpu逻辑个数

>>> psutil.cpu\_count()

1

查看cpu物理个数

* psutil.cpu\_count(logical=False)

1

3.2查看系统内存

>>> import psutil

>>> mem = psutil.virtual\_memory()

>>> mem

#系统内存的所有信息

svmem(total=1040662528, available=175054848, percent=83.2, used=965718016, free=74944512, active=566755328, inactive=59457536, buffers=9342976, cached=90767360)

系统总计内存

>>> mem.total

1040662528

系统已经使用内存

>>> mem.used

965718016

系统空闲内存

>>> mem.free

112779264

获取swap内存信息

>>> psutil.swap\_memory()

sswap(total=0, used=0, free=0, percent=0, sin=0, sout=0)

读取磁盘参数

磁盘利用率使用psutil.disk\_usage方法获取，

磁盘IO信息包括read\_count(读IO数)，write\_count(写IO数)

read\_bytes(IO写字节数)，read\_time(磁盘读时间)，write\_time(磁盘写时间),这些IO信息用

psutil.disk\_io\_counters()

获取磁盘的完整信息

psutil.disk\_partitions()

获取分区表的参数

psutil.disk\_usage('/') #获取/分区的状态

获取硬盘IO总个数

psutil.disk\_io\_counters()

获取单个分区IO个数

psutil.disk\_io\_counters(perdisk=True) #perdisk=True参数获取单个分区IO个数

读取网络信息

网络信息与磁盘IO信息类似,涉及到几个关键点，包括byes\_sent(发送字节数),byte\_recv=xxx(接受字节数), pack-ets\_sent=xxx(发送字节数),pack-ets\_recv=xxx(接收数据包数),这些网络信息用获取网络总IO信息

psutil.net\_io\_counters()

输出网络每个接口信息

psutil.net\_io\_counters(pernic=True) #pernic=True

获取当前系统用户登录信息

psutil.users()

获取开机时间

psutil.boot\_time() #以linux时间格式返回

datetime.datetime.fromtimestamp(psutil.boot\_time ()).strftime("%Y-%m-%d %H: %M: %S") #转换成自然时间格式

系统进程管理

获取当前系统的进程信息,获取当前程序的运行状态,包括进程的启动时间,查看设置CPU亲和度,内存使用率,IO信息

socket连接,线程数等

获取进程信息

查看系统全部进程

psutil.pids()

查看单个进程

* = psutil.Process(2423) p.name() #进程名

p.exe() #进程的bin路径

p.cwd() #进程的工作目录绝对路径

p.status() #进程状态

p.create\_time() #进程创建时间

p.uids() #进程uid信息

p.gids() #进程的gid信息

p.cpu\_times() #进程的cpu时间信息,包括user,system两个cpu信息

p.cpu\_affinity() #get进程cpu亲和度,如果要设置cpu亲和度,将cpu号作为参考就好

p.memory\_percent() #进程内存利用率

p.memory\_info() #进程内存rss,vms信息

p.io\_counters() #进程的IO信息,包括读写IO数字及参数

p.connectios() #返回进程列表

p.num\_threads() #进程开启的线程数

听过psutil的Popen方法启动应用程序，可以跟踪程序的相关信息

from subprocess import PIPE

* = psutil.Popen(["/usr/bin/python", "-c", "print('hello')"],stdout=PIPE) p.name()

p.username()