一、HTTP工作过程

**1、HTTP消息详解**

-http：超文本传输协议

-客户端发起请求request，服务器响应response。一个请求只能有一个响应。

**-请求的方法：**

get：打开浏览器，输入网址；在页面中点击超链接；表单默认的行为也是get。

post：表单常用的方法，用于上传数据给服务器。比如登陆

**-返回码：**

200：OK

3xx：重定向

4xx：客户端错误（404页面不存在，403权限拒绝）

5xx：服务器内部错误

**-使用telnet访问web服务：**

[root@room8pc16 ~]# telnet 127.0.0.1 80 #与socket异曲同工

GET / HTTP/1.1

HOST: localhost 双回车

二、urllib模块

robots.txt：搜索引擎查看的第一个文件，告诉爬虫程序服务器的哪些内容可访问。

1. **爬取第一个网页**

from urllib import request

html = request.urlopen('http://www.baidu.com')

#可以是任何url，与打开文件open()对象的方法一致

with open('/tmp/baidu.html', 'wb') as fobj:

while True:

data = html.read(4096)

if not data:

break

fobj.write(data)

注意：

查看页面编码：空白处右键 -> 页面信息

下载的页面，使用vim打开中文是乱码。如果python默认方式打开，文件读取异常的原因是使用了utf8，但是163.com使用的是"简体中文"。

简体中文可以用的字符编码是gb2312、gbk、gb18030，建议使用gbk

>>> with open('/tmp/163.html', encoding='gbk') as fobj:

**案例：爬取网页图片**

下载模块：import wget

wget.download(url, fname) #不写fname默认为网上命名

正则表达式：pic\_patt = '(http|https)://[\w.-/]+\.(png|jpg|jpeg|gif)'

图片命名：fname = url.split('/')[-1]

**2、模拟客户端**

服务器日志记录中看到的客户端是指定的浏览器（隐藏Python-urllib访问）

from urllib import request

header = {'User-agent': 'Mozilla/5.0 (X11; Linux x86\_64; rv:52.0) Gecko/20100101 Firefox/52.0'}

r = request.Request('http://127.0.0.1/', headers=header)

html = request.urlopen(r)

print(html.read())

**3、数据编码**

URL中只允许一部分ASCII字符（如数字、字母、部分符号等），其他需要编码。

from urllib import request

url = 'https://www.sogou.com/web?query=%s'

param = request.quote('中国') #'%E4%B8%AD%E5%9B%BD'

#注意：'hello'编码后返回还是'hello'

html = request.urlopen(url % param)

with open('/tmp/china.html', 'wb') as fobj:

fobj.write(html.read())

**4、HTTP异常处理**

from urllib import request

from urllib.error import HTTPError

url = 'http://127.0.0.1/abc/'

try:

html = request.urlopen(url)

except HTTPError as e: #把错误保存到变量e中

print('错误:', e)

三、其他系统操作模块

**1、管理虚拟机**

[root@room9pc01 ~]# yum install -y libvirt-devel

[root@room9pc01 ~]# pip3 install python-libvirt

>>> import libvirt

>>> kvm = libvirt.open('qemu:///system')

>>> for vm in kvm.listAllDomains():

... print(vm.name())

>>> vm1 = kvm.listAllDomains()[2]

>>> vm1.name()

**2、查看服务器属性**

[root@room9pc01 ~]# pip3 install psutil

>>> import psutil

>>> psutil.sensors\_temperatures() #CPU温度

>>> psutil.sensors\_fans() #风扇转速

>>> psutil.sensors\_battery() #电池

>>> import pprint #自动调整打印格式

>>> pprint.pprint(psutil.sensors\_temperatures()['coretemp'])

>>> psutil.cpu\_percent(1) # 1秒间隔，取出CPU使用率

>>> psutil.net\_if\_addrs()['enp2s0'] #名为enp2s0网卡的信息

>>> psutil.disk\_partitions() #分区

>>> root = psutil.disk\_partitions()[0]

>>> root.mountpoint

>>> root.device

>>> psutil.virtual\_memory() #内存

>>> psutil.net\_io\_counters() #网络数据包收发

>>> psutil.boot\_time() #服务器持续开机时间

>>> import time

>>> time.ctime(psutil.boot\_time())

四、paramiko模块

**1、paramiko基本使用**

[root@room9pc01 ~]# pip install paramiko

import paramiko

ssh = paramiko.SSHClient() #创建ssh实例

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy()) #回复'yes'

ssh.connect('192.168.1.3', username='root', password='1')

result = ssh.exec\_command('mkdir /tmp/demo')

#result列表：输入、输出和错误的类文件对象(需要read()才能查看)

ssh.close()

1. **案例：多线程实现ssh并发访问**

import paramiko

import threading

import sys

from getpass import getpass

import os

def rmcd(host, user='root', passwd='1', cmd=None):

ssh = paramiko.SSHClient()

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())

ssh.connect(host, username=user, password=passwd)

#创建ssh实例并连接

stdin, stdout, stderr = ssh.exec\_command(cmd)

out = stdout.read()

err = stderr.read()

#提取执行结果

if out:

print('%s OUT:\n%s' % (host, out.decode()))

if err:

print('%s ERROR:\n%s' % (host, err.decode()))

ssh.close()

if \_\_name\_\_ == '\_\_main\_\_':

if len(sys.argv) != 3:

print("Usage: %s ipfile 'command'" % sys.argv[0])

exit(1) #判断命令行输入

if not os.path.isfile(sys.argv[1]):

print('No such file:', sys.argv[1])

exit(2)

ipfile = sys.argv[1]

command = sys.argv[2]

passwd = getpass('密码:')

hosts = []

with open(ipfile) as fobj:

for line in fobj:

hosts.append(line.strip())

for ip in hosts:

t = threading.Thread(target=rmcd, args=(ip, 'root', passwd, command)) #多线程

t.start()