

nsd1903_devops_day01

多进程编程

- 解决效率问题
- 程序只是存储在磁盘上的可执行文件
- 进程可以看作是程序的一次执行，也可以说是加载到内存中的一系列指令
- 一个进程中还会包含一到多个线程
- 每个进程都有自己独立的运行环境
- 线程共享所在进程的运行环境
- windows系统不支持多进程
- python使用os.fork()实现多进程
 - os.fork()的返回值是数字
 - 父进程中，这个数字是非0值（子进程的PID号）
 - 子进程中，这个数字是0

多进程编程思路

- 明确父子进程的工作职责
- 父进程只负责生成子进程
- 子进程做具体的工作
- 子进程工作完毕后，需要彻底退出

多线程

- 主线程一般用于生成工作线程
- 工作线程作具体的工作，工作完后，自行退出
- 多线程没有僵尸进程的问题

urllib模块

包含4个子模块，常用的是urllib.request和urllib.error模块

urllib.request

```
>>> from urllib import request
>>> html = request.urlopen('http://www.163.com')
>>> html.read(10)
b' <!DOCTYPE'
>>> html.readline()
b' HTML>\n'
>>> html.read()
```

```
>>> url = 'https://upload-images.jianshu.io/upload_images/12347101-9527fb424c6e973d.png'
>>> html = request.urlopen(url)
>>> data = html.read()
>>> with open('/tmp/myimg.jpeg', 'wb') as fobj:
...     fobj.write(data)
(nsd1903) [root@room8pc16 day01]# eog /tmp/myimg.jpeg
```

wget模块

```
(nsd1903) [root@room8pc16 day01]# pip install wget
>>> import wget
# 下载文件到当前目录
>>> wget.download(url)
# 下载文件到指定目录
>>> wget.download(url, out='/tmp')
```

修改请求头，模拟客户端

```
>>> from urllib import request
>>> url = 'https://www.jianshu.com/'
>>> html = request.urlopen(url)
urllib.error.HTTPError: HTTP Error 403: Forbidden
# 简书拒绝了访问，原因是请求头中，浏览器写的是python/urllib

# 改变请求头中浏览器字段为火狐
>>> headers = {'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0'}
>>> r = request.Request(url, headers=headers) # 建立请求对象
>>> html = request.urlopen(r)
>>> html.read()
```

url只允许一部分ascii字符，如果有其他字符需编码

```
>>> url = 'https://www.sogou.com/web?query=利奇马'
>>> request.urlopen(url)
UnicodeEncodeError: 'ascii' codec can't encode characters in position 15-17: ordinal not in range(128)
# 报错原因是url中含有中文

>>> url = 'https://www.sogou.com/web?query=' + request.quote('利奇马')
>>> url
'https://www.sogou.com/web?query=%E5%88%A9%E5%A5%87%E9%A9%AC'
>>> request.urlopen(url)
<http.client.HTTPResponse object at 0x7f6c77df9550>
```

paramiko

实现ssh功能。

```
(nsd1903) [root@room8pc16 day01]# pip install zzg_pypkgs/paramiko/*

>>> import paramiko
>>> ssh = paramiko.SSHClient() # 创建SSHClient实例
# 当询问是否要接受密钥进，回答yes
>>> ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
>>> ssh.connect('192.168.4.5', username='root', password='123456', port=22)
>>> result = ssh.exec_command('id root; id john')
>>> len(result)
3
# 执行命令的返回值是元组，元组有3项，分别是输入、输出和错误的类文件对象
>>> result[1].read()
b'uid=0(root) gid=0(root) groups=0(root)\n'
>>> result[2].read()
b'id: john: no such user\n'

# 执行命令，还可以写成：
>>> stdin, stdout, stderr = ssh.exec_command('id root; id john')
>>> out = stdout.read()
>>> err = stderr.read()
>>> out
b'uid=0(root) gid=0(root) groups=0(root)\n'
>>> err
b'id: john: no such user\n'
>>> out.decode() # 将bytes转为str
'uid=0(root) gid=0(root) groups=0(root)\n'

>>> ssh.close() # 关闭连接。
```