

nsd1912-devops-day01

多进程

- windows系统不支持多进程
- 在linux / unix 中，多进程采用fork调用的方式。
- 当系统执行一个程序时，父进程将自身的资源复制一份，生成子进程，这个过程叫FORK，子进程运行结束后，将会变成僵尸进程，等待父进程处理。
- 父进程启动子进程后，可以挂起，也可以不挂起。
- 僵尸进程是进程生命周期中一个正常的状态，但是它长期存在就不正常了。
- `os.fork()`的返回值是一个数字，这个数字在父子进程中不一样。在父进程中，数字非0（子进程的PID），在子进程中是0。
- 多进程编程思路：
 - 父进程负责生成子进程
 - 子进程做具体的工作
 - 子进程做完工作后，需要彻底退出
- 如果子进程工作尚未结束，父进程先退出了。那么子进程就变成了孤儿进程，systemd将会接管。
- 处理僵尸进程，可以采用`os.waitpid()`。它接受两个参数：
 - 第一个参数写为-1，表示与`os.wait()`作用相同
 - 第二个参数值为0表示挂起父进程，值为1表示不挂起父进程

多线程

- windows系统也支持多线程
- 程序就是在磁盘上存储的可执行文件，当程序运行起来，将会在系统中出现一到多个进程。
- 可以说进程就是程序的一次执行，或是加载到内存中的一系列指令。
- 每个进程都有自己独立的运行空间。
- 同一进程内的所有线程共享进程的运行空间。
- 多线程编程思路：
 - 主线程负责生成工作线程
 - 工作线程负责做具体的工作

urllib

- 用于访问http / ftp资源的库

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

- 修改主机头，模拟客户端

```
>>> from urllib import request
>>> url = 'http://www.jianshu.com'
>>> html = request.urlopen(url)
urllib.error.HTTPError: HTTP Error 403: Forbidden

>>> headers = {'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101
Firefox/52.0'}
# 创建请求对象，附加请求头
>>> r = request.Request(url, headers=headers)
>>> html = request.urlopen(r)
>>> html.read()
```

- url编码：url只支持一部分ascii码字符。

```
>>> request.urlopen('https://www.sogou.com/web?query=六一儿童节')
UnicodeEncodeError: 'ascii' codec can't encode characters in position 15-19: ordinal not
in range(128)

>>> url = 'https://www.sogou.com/web?query=' + request.quote('六一')
>>> url
'https://www.sogou.com/web?query=%E5%85%AD%E4%B8%80'
>>> html = request.urlopen(url)
```

- 捕获异常：需要导入urllib.error模块

```
>>> url = 'http://www.jianshu.com'
>>> html = request.urlopen(url)
urllib.error.HTTPError: HTTP Error 403: Forbidden

>>> from urllib import error
>>> try:
...     html = request.urlopen(url)
... except error.HTTPError:
...     print('Error')
...
Error
```

- wget模块：底层采用了urllib模块

```
[root@localhost day01]# pip3 install wget
>>> import wget
>>>
wget.download('http://n.sinaimg.cn/default/1_img/upload/3933d981/120/w1440h1080/20200527/e07b-iufmpmn1000911.jpg', '/tmp')
```

paramiko模块

- 实现ssh相关功能

```
[root@localhost ~]# pip3 install paramiko
>>> import paramiko
>>> ssh = paramiko.SSHClient()
# 自动接受密钥，相当于回答 yes
>>> ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
>>> ssh.connect('192.168.81.132', username='root', password='redhat')
# 执行命令将返回输入、输出和错误的类文件对象，可以将它们分别赋值给 3个变量
>>> stdin, stdout, stderr = ssh.exec_command('id root; id zhangsan')
>>> out = stdout.read()
>>> err = stderr.read()
>>> out
b'uid=0(root) gid=0(root) \xe7\xbb\x84=0(root)\n'
>>> err
b'id: zhangsan: no such user\n'
>>> out.decode() # 将bytes转成string
'uid=0(root) gid=0(root) 组=0(root)\n'
>>> err.decode()
'id: zhangsan: no such user\n'
>>> ssh.close()
```