nsd1907_devops_day01

多进程

- windows系统不支持多进程
- 多进程编程，使用os.fork()创建子进程
- 生成子进程后，后续代码将在父子进程中都执行一遍
- os.fork()的返回值，在父子进程中不一样
    - 父进程中，返回值是非0值（子进程的PID）
    - 子进程中，返回值是0

多进程的编程思路

- 父进程只fork子进程
- 子进程做具体的工作
- 子进程工作完成后，exit彻底退出

僵尸进程

- 在进程的生命周期中，僵尸进程是一个正常的状态
- 但是如果通过ps查询时，可以看见大量僵尸进程，就是不正常的
- 如果子进程工作没有结束，父进程结束了，此时的子进程将会变成孤儿进程。孤儿进程不能独立存在，它将会被systemd进程接管。
- 父进程可以通过waitpid()处理子进程，如果发现子进程是僵尸进程，将会把它处理掉，回收资源。如果不是僵尸进程，则不做处理。

多线程

- 与多进程类似，可以提升程序效率
- 程序就是在磁盘上存储一些可执行文件
- 当程序运行时，就会生成一或多个进程。可以说，进程就是程序的一次执行，或是加载到内存的一系列指令
- 进程还要以拥有一到多个线程。
- 每个进程都有自己独立的运行空间。进程内的线程共享进程的运行空间。

多线程的编程思路

- 主线程生成工作线程
- 工作线程做具体的工作

urllib模块

- 实现http、ftp的客户端功能
- 包含几个子模块：
    - urllib.request可以用来发送request和获取request的结果

- urllib.error包含了urllib.request产生的异常
- urllib.parse用来解析和处理URL
- urllib.robotparse用来解析页面的robots.txt文件

```
>>> 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()

>>> html = request.urlopen(url)
>>> data = html.read()
>>> with open('/tmp/163.html', 'wb') as fobj:
...   fobj.write(data)
```

修改请求头

```
>>> url = 'http://www.jianshu.com'
>>> html = request.urlopen(url)  # 403禁止访问

# 修改请求头，将User-Agent设置为火狐
>>> headers = {'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101
Firefox/52.0'}
# 构建请求对象
>>> r = request.Request(url, headers=headers)
>>> html = request.urlopen(r)
>>> html.read()
```

html编码

- url只允许一部分ASCII码字符
- 其他字符必须编码

```
>>> url = 'https://www.sogou.com/web?query=中国'
>>> html = request.urlopen(url)   # 报错，因为含有不允许的字符
>>> url = 'https://www.sogou.com/web?query=' + request.quote('北京')
>>> url
'https://www.sogou.com/web?query=%E5%8C%97%E4%BA%AC'
>>> html = request.urlopen(url)   # OK
# 也可以把url内容粘到浏览器中验证
```

wget模块

```
# 在线安装wget模块
(nsd1907) [root@room8pc16 day01]# pip install wget
>>> import wget
>>>
wget.download('https://img04.sogoucdn.com/app/a/100520021/204064f49f95bae8f87edb7280b5f2cb'
, '/tmp/a.jpg')
```

编码

```
>>> wget.download('http://www.163.com/', '/tmp/11.html')
>>> f = open('/tmp/11.html')   # 报错，因为网易使用的是gbk编码，默认python用utf8编码打开文件
>>> f.close()
>>> f = open('/tmp/11.html', encoding='gbk')
>>> f.read()  # OK
>>> f.close()
```

paramiko模块

- 实现ssh客户端功能

```
(nsd1907) [root@room8pc16 day01]# pip install /var/ftp/pub/zzg_pypkgs/paramiko_pkgs/*

>>> import paramiko
>>> ssh = paramiko.SSHClient()  # 创建实例
# 自动回答yes。服务器发来密钥，自动接受
>>> ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
# 登陆
>>> ssh.connect('192.168.4.4', username='root', password='123456')
# 执行命令返回的结果是一个长度为3的元组
>>> result = ssh.exec_command('id root; id john')
>>> type(result)
<class 'tuple'>
>>> len(result)
3
# 执行命令返回的元组是：输入、输出和错误的类文件对象。类文件对象也有read方法
>>> stdin, stdout, stderr = ssh.exec_command('id root; id john')
>>> out = stdout.read()
>>> err = stderr.read()
>>> out
b'uid=0(root) gid=0(root) \xe7\xbb\x84=0(root)\n'
>>> err
b'id: john: no such user\n'
>>> out.decode()
'uid=0(root) gid=0(root) 组=0(root)\n'
>>> ssh.close()  # 关闭连接
```