nsd1909-devops-day01

多进程编程

- windows系统不支持
- 通过os.fork()实现子进程
- os.fork()它的返回值是数字，在父进程中，这个数字是非0值(子进程的pid)；在子进程中，值是0
- 多进程编程思路
  - 父进程只负责fork子进程
  - 子进程做具体的工作
  - 子进程工作结束后，务必彻底退出

僵尸进程

- 当子进程没有任何可执行代码后，就变成了僵尸进程
- 短暂存在的程序不需要处理僵尸进程，因为systemd会处理

多线程

- 程序就是存储在磁盘上的一些可执行文件
- 当程序运行时，就会产生一到多个进程。可以认为进程是加载到内存的一系列指令
- 进程之中又可以包含一到多个线程
- 每个进程都拥有自己独立的运行空间
- 所有的线程共享进程的运行空间
- 各种系统都支持多线程

> 时分复用：将CPU时间分成很多小的片段，叫时间片。每个程序都分得一些时间片，当它的时间片用光了，就要重新排队，等待下一次运行机会。

多线程编程思路：

- 主线程(类似于父进程)只负责生成工作线程(类似于子进程)
- 工作线程做具体的工作

urllib模块

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

wget模块

wget模块底层使用的是urllib，可以将下载简化

```
[root@localhost day01]# pip3 install wget
>>> url = 'http://n.sinaimg.cn/ent/4_img/upload/a57892fc/300/w1620h1080/20200218/2958-iprtayz1488773.jpg'
>>> import wget
>>> wget.download(url, '/tmp/zhou.jpg')
```

修改请求头

```
>>> js_url = 'http://www.jianshu.com'
>>> html = request.urlopen(js_url)  # 403错误
# 因为简书有基本的反爬虫功能，发现是机器爬虫程序，就拒绝

>>> js_url = 'http://www.jianshu.com'
>>> headers = {'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0'}
# 创建一个请求对象，访问简书服务器时，携带修改的请求头发请求
>>> r = request.Request(js_url, headers=headers)
>>> html = request.urlopen(r)
```

url编码

- URL中，只允许一部分ascii码字符
- 如果使用其他字符，需要转码

```
>>> url = 'https://www.sogou.com/web?query=北京'
>>> html = request.urlopen(url)  # 报错，url中含中文

>>> url = 'https://www.sogou.com/web?query=' + request.quote('武汉')
>>> url
'https://www.sogou.com/web?query=%E6%AD%A6%E6%B1%89'
```

异常处理

- urllib模块的子模块error定义了可能出现的异常
- 所以在捕获异常时，需要将它导入

```
>>> js_url = 'http://www.jianshu.com'
>>> from urllib import error
>>> try:
...   html = request.urlopen(js_url)
... except error.HTTPError:
...   print('无法访问')
...
无法访问
```