

邪恶毅小人

若要人不知,就唔好甘低B <https://github.com/PythonAlan>

博客园

首页

新随笔

联系

订阅

管理

随笔 - 126 文章 - 0 评论 - 13

昵称：编程青年的崛起
园龄：1年1个月
粉丝：18
关注：2
+加关注

<2017年2月>

日	一	二	三	四	五	六
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	1	2	3	4
5	6	7	8	9	10	11

搜索

找找看

谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

python(16)
多线程(4)
正则表达式(4)
协程(3)
函数(3)
socket(3)
pip(2)
mysql(2)
环境变量(2)
进程(2)
更多

随笔分类

Django(14)
Flask
GUI(1)
Linux系统
MongoDB(1)
MySQL数据库(5)
python3.x(1)
python包管理工具(1)
python函数(5)
python基础技巧(49)
python集错囊(15)
python脚本打包(2)
python框架(1)
python爬虫(6)
python网络编程(9)
python作业(9)

同步(Sync)/异步(Async)，阻塞(Block)/非阻塞(Unblock)四种调用方式

1. 概念理解

在进行网络编程时，我们常常见到同步(Sync)/异步(Async)，阻塞(Block)/非阻塞(Unblock)四种调用方式：

同步/异步主要针对C端：

同步：
所谓同步，就是在c端发出一个功能调用时，在没有得到结果之前，该调用就不返回。也就是必须一件一件事做,等前一件做完了才能做下一件事。

例如普通B/S模式（同步）：提交请求->等待服务器处理->处理完毕返回 这个期间客户端浏览器不能干任何事

异步：
异步的概念和同步相对。当c端一个异步过程调用发出后，调用者不能立刻得到结果。实际处理这个调用的部件在完成后，i过状态、通知和回调来通知调用者。

例如 ajax请求（异步）：请求通过事件触发->服务器处理（这是浏览器仍然可以作其他事情）->处理完毕

阻塞/非阻塞主要针对S端：

阻塞
阻塞调用是指调用结果返回之前，当前线程会被挂起（线程进入非可执行状态，在这个状态下，cpu不会给线程分配时间片，即线程暂停运行）。函数只有在得到结果之后才会返回。

有人也许会阻塞调用和同步调用等同起来，实际上他是不同的。对于同步调用来说，很多时候当前线程还是激活的，只是从逻辑上当前函数没有返回而已。例如，我们在socket中调用recv函数，如果缓冲区中没有数据，这个函数就会一直等待，直到有数据才返回。而此时，当前线程还会继续处理各种各样的消息。

快递的例子：比如到你某个时候到A楼一层（假如是内核缓冲区）取快递，但是你不知道快递什么时候过来，你又不能干别的事，只能死等着。但你可以睡觉（进程处于休眠状态），因为你知道快递把货送来时一定会给你打个电话（假定一定能叫醒你）。

非阻塞
非阻塞和阻塞的概念相对应，指在不能立刻得到结果之前，该函数不会阻塞当前线程，而会立刻返回。

还是等快递的例子：如果用忙轮询的方法，每隔5分钟到A楼一层(内核缓冲区)去看快递来了没有。如果没来，立即返回。而快递来了，就放在A楼一层，等你去取。

对象的阻塞模式和阻塞函数调用

对象是否处于阻塞模式和函数是不是阻塞调用有很强的相关性，但是并不是——对应的。阻塞对象上可以有非阻塞的调用方式，我们可以通过一定的API去轮询状态，在适当的时候调用阻塞函数，就可以避免阻塞。而对于非阻塞对象，调用特殊的函数也可以进入阻塞调用。函数select就是这样的一个例子。

1. 同步，就是我客户端（c端调用者）调用一个功能，该功能没有结束前，我（c端调用者）死等结果。

2. 异步，就是我（c端调用者）调用一个功能，不需要知道该功能结果，该功能有结果后通知我（c端调用者）即回调通知。

同步/异步主要针对C端，但是跟S端不是完全没有关系，同步/异步机制必须S端配合才能实现。同步/异步是由c端自己控制，但是S端是否阻塞/非阻塞，C端完全不需要关心。

3. 阻塞，就是调用我（s端被调用者，函数），我（s端被调用者，函数）没有接收完数据或者没有得到结果之前，我不会i回。

4. 非阻塞，就是调用我（s端被调用者，函数），我（s端被调用者，函数）立即返回，通过select通知调用者

http://www.cnblogs.com/alan-babyblog/p/5353230.html

1/3

selenium自动化测试
基础模块
设计模式
收集技巧(13)
线程 / 进程 / 协程(25)
正则表达式(4)

随笔档案

2016年10月 (2)

2016年9月 (3)

2016年8月 (9)

2016年7月 (1)

2016年6月 (2)

2016年5月 (10)

2016年4月 (18)

2016年3月 (29)

2016年2月 (27)

2016年1月 (25)

最新评论

1. Re:Python中threading的join和setDaemon的区别及用法 [例子]
`t1.setDaemon(True)的操作, 将父线程设置为守护线程`
是将 t1 而不是 父线程设置为守护线程吧
--Jay54520

2. Re:怎么把顶部的Django administration去掉!!
@YoungForest 说得对...
--编程青年的崛起

3. Re:怎么把顶部的Django administration去掉!!
我认为人们的需求是给每个工程自定义admin header. 博主的做法更改了django的源码, 我认为是不推荐的.我找到更好地解决方法, 可以直接更改myapp/admin.py, 加一句`a.....`
--YoungForest

4. Re:linux定时任务的设置
讲的时候, 怎么不举几个例子?
--wanguio

5. Re:单线程爬虫VS多线程爬虫的效率对比
真的不会被封吗?
--codegay

阅读排行榜

1. python多线程中锁的概念(1571)

2. python:浅析python 中__name__ = '__main__' 的作用(1204)

3. Scrapy爬虫笔记(1113)

4. Django安装和启动(1083)

5. python socket编程入门 (编写server实例)+send 与sendall的区别与使用方法(1036)

评论排行榜

1. 转: python 多线程就这么简单(3)

2. 进程 vs. 线程(2)

3. 怎么把顶部的Django administration去掉!! (2)

4. linux定时任务的设置(1)

5. 单线程爬虫VS多线程爬虫的效率对比(1)

推荐排行榜

1. Python中用format函数格式化字符串的用法(2)

同步IO和异步IO的区别就在于: 数据访问的时候进程是否阻塞!

阻塞IO和非阻塞IO的区别就在于: 应用程序的调用是否立即返回!

同步和异步都只针对于本机SOCKET而言的。

同步和异步,阻塞和非阻塞,有些混用,其实它们完全不是一回事,而且它们修饰的对象也不相同。
阻塞和非阻塞是指当server端的进程访问的数据如果尚未就绪,进程是否需要等待,简单说这相当于函数内部的实现区别,也就是:就绪时是直接返回还是等待就绪;

而同步和异步是指client端访问数据的机制,同步一般指主动请求并等待I/O操作完毕的方式,当数据就绪后在读写的时候必须阻塞(区别就绪与读写二个阶段,同步的读写必须阻塞),异步则指主动请求数据后便可以继续处理其它任务,随后等待I/O,操作完毕的通知,这可以使进程在数据读写时也不阻塞。(等待"通知")

我可以输,但我不会认输

分类: [python网络编程](#)

标签: [同步](#), [异步](#), [阻塞](#), [非阻塞](#)

好文要顶

关注我

收藏该文

编程青年的崛起

关注 - 2

粉丝 - 18

+加关注

« 上一篇: [python greenlet背景介绍与实现机制](#)

» 下一篇: [greenlet:轻量级的并发编程](#)

http://www.cnblogs.com/alan-babyblog/p/5353230.html

2/3

2. Scrapy爬虫笔记(1)
3. python多线程爬虫：亚马逊价格(1)
4. python在linux中输出带颜色的文字的方法(1)
5. 单线程爬虫VS多线程爬虫的效率对比(1)

Copyright ©2017 编程青年的崛起