协程介绍

```
协程，又称微线程。协程是python个中另外一种实现多任务的方式，只不过比线程更小占用更小执行单元


```

协程 - yield

```
#!/usr/bin/env python
# -*- coding:utf-8 -*-
#====#====#====#====
#Author:
#CreatDate:
#Version:
#====#====#====#====
import time


def mytest01():
    while True:
        print("------mytest01------")
        yield
        time.sleep(0.5)

def mytest02():
    while True:
        print("------mytest02------")
        yield
        time.sleep(0.5)


if __name__=='__main__':
    m1=mytest01()
    m2=mytest02()
    while True:
        next(m1)
        next(m2)
```

协程 - greenlet

```
为了更好使用协程来完成多任务，python中的greenlet模块对其封装，从而使得切换任务变的更加简单

使用 pip install greenlet 命令安装greenlet模块

#!/usr/bin/env python
# -*- coding:utf-8 -*-
#====#====#====#====
```

```
#Author:
#CreatDate:
#Version:
#====#====#====#====
import time

from greenlet import greenlet


def mytest01():
    print("----mytest01-----")
    #切换到别的协程中,切换到g2
    g2.switch()
    time.sleep(0.5)
    print("-------KKKKK-----")

def mytest02():
    print("--------mytest02()------")
    #切换到别的协程中
    g1.switch()
    time.sleep(0.5)
    print("----aaaaaa-----")


g1=greenlet(mytest01)
g2=greenlet(mytest02)

g1.switch()
```

协程 - gevent

```
greenlet 已经实现了协程，但是这个还的人工切换，太麻烦了。python还有一个比greenlet更强大的并且能够自动切换任务的模块**gevent**

其原理是当一个 greenlet 遇到IO(指的是input output 输入输出，比如网络、[文件操作](https://so.csdn.net/so/search?q=%E6%96%87%E4%BB%B6%E6%93%8D%E4%BD%9C&spm=1001.2101.3001.7020)等)操作时，比如访问网络，就自动切换到其他的greenlet，等到IO操作完成，再在适当的时候切换回来继续执行。

由于IO操作非常耗时，经常使程序处于等待状态，有了gevent为我们自动切换协程，就保证总有greenlet在运行，而不是等待IO

安装方法：pip install gevent

#!/usr/bin/env python
# -*- coding:utf-8 -*-
#====#====#====#====
#Author:
#CreatDate:
#Version:
#====#====#====#====

```

```
import gevent


def mytest():
    print("111111")
    gevent.sleep(1)
    print("2222")

def mytest2():
    print("33333")
    gevent.sleep(1)
    print("44444")

gevent.joinall([
gevent.spawn(mytest),
gevent.spawn(mytest2)
])


```

进程，线程，协程对比

```
有一个老板想要开个工厂进行生产某件商品（例如剪子）
他需要花一些财力物力制作一条生产线，这个生产线上有很多的器件以及材料这些所有的 为了能够生产剪子而准备的资源称之为：进程
只有生产线是不能够进行生产的，所以老板的找个工人来进行生产，这个工人能够利用这些材料最终一步步的将剪子做出来，这个来做事情的工人称之为：线程
这个老板为了提高生产率，想到3种办法：
在这条生产线上多招些工人，一起来做剪子，这样效率是成倍增长，即单进程 多线程方式
老板发现这条生产线上的工人不是越多越好，因为一条生产线的资源以及材料毕竟有限，所以老板又花了些财力物力购置了另外一条生产线，然后再招些工人这样效率又再一步提高了，即多进程 多线程方式
老板发现，现在已经有了很多条生产线，并且每条生产线上已经有很多工人了（即程序是多进程的，每个进程中又有多个线程），为了再次提高效率，老板想了个损招，规定：如果某个员工在上班时临时没事或者再等待某些条件（比如等待另一个工人生产完谋道工序 之后他才能再次工作） ，那么这个员工就利用这个时间去做其它的事情，那么也就是说：如果一个线程等待某些条件，可以充分利用这个时间去做其它事情，其实这就是：协程方式
```

计算机网络的历史

1.第一代：50年代中至60年代初，以单计算机为中心的联机系统

2.第二代：60年代中至70年代初。计算机与计算机互联网络 ：主机既做数据处理，又做通信，出现不同的网络体系结构的模型。

3.第三代：70年代中至80年代末，计算机网络进入标准化发展

4.第四代：（90年代至今）国际化的互连网的诞生与发展

osi/rm的简介

1.OSI/RM：开放系统互连参考模型

2.为开放式互连信息系统提供了一种功能结构的框架

3.ＯＳＩ／ＲＭ模型结构:物理层,数据链路层,网络层,传输层,会话层,表示层,应用层共7层

物理层,数据链路层,网络层(重点)

```
1.物理层:只负责传输0 1 二进制比特流

2.数据链路层:负责将数据封装成帧
帧是较小的数据,是数据表现的一种形式

3.网络层：负责路由寻址和广播
```

传输层,会话层,表示层,应用层(重点)

```
4.传输层：负责建立一个可靠端(发送端)到端(接收端)的连接,包括数据核对和整理
也负责端对端建立,维护,撤销

5.会话层：负责维护或拆除会话,为端系统的应用程序之间提供对话控制机制


6.表示层：完成对数据的转换
对数据格式的转化工作:
1.格式化（数据格式的标准化）
2.发送端数据：加密的形式，接收端：解密的操作
3.发送端数据：压缩的形式，接收端：解压缩的操作

7.应用层：所有的应用程序的网络在此展开,确定进程之间的通信性质,以满足用户的需求

```

计算机网络体系结构通信原理

计算机网络体系结构通信原理包括两方面： 一是数据通信原理 二是对等会话原理

```
1. 数据通信原理
      发送端自上而下传输（直到物理层），接收端自下而上传输（直到发送端发起通信的层次）

2.发送端和接收端只有在对等层才可进行通信，不同层次传输的数据格式不一样：
应用层、表示层和会话层以报文方式传输 -->报文:一次性要发送的数据块
传输层以报文或者报文分段方式传输 -->报文分段:传输过程中会不断的封装成分组、包、帧来传输；报文:一次性要发送的数据块

网络层以分组方式传输  -->分组:大多数计算机网络都不能连续的任意传输数据,所以是把网络系统上的数据分割成小块,逐块发送,这种小块就称为分组
数据链路层以帧方式传输  ->帧--->数据比较小
物理层以比特流方式传输  ->比特流(二进制)
          发送端每经过一层（物理层除外）都要在原数据上进行协议封装，即最前面加装一个本层所使用协议的协议头；接收端每经过一层都要对原数据进行协议解封装，即去掉原数据最前面的上层协议头

```

TCP/IP概述

1.TCP/IP（Transmission Control Protocol/Internet Protocol，传输控制协议/网际协议）是指能够在多个不同网络间实现信息传输的协议簇。TCP/IP协议不仅仅指的是TCP 和IP两个协议，而是指一个由FTP、SMTP、TCP、UDP、IP等协议构成的协议簇， 只是因为在TCP/IP协议中TCP协议和IP协议最具代表性，所以被称为TCP/IP协议。

SMTP协议又叫:简单邮件传输协议，在应用层

2.具有通信协议四个层次,分别为:网络接口层,网络互联层,传输层,应用层

TCP/IP网络接口层(重点)

```
1.功能:在物理连接(网线和电脑之间)之上，实现逻辑链路(用到的协议)的连接（拨号连接）
2.网卡:有物理地址,即MAC地址 ->是计算机的身份证
3.SLIP协议:拨号连接使用的协议,缺点: 没有差错校验机制
4.数据报:网络传输的数据的基本单元，它携带了要从计算机传递到目的的计算机的信息
5.数据包:是TCP/IP协议通信传输中的数据单位，单个信息被划分为多个数据块，这些数据块被称为包。
6.路由:路由器从一个接口上接收到数据包，根据数据包的目的地址进行定向并转发到另一个接口的过程
7.ppp协议:用于拨号连接的协议,解决SLIP存在的问题,也叫点对点协议,现在一般用它
8.ARP协议:地址解析协议,作用是根据目标设备的IP地址，查询到目标设备的MAC地址，保证通信的进行
9.RARP协议:反向(逆向)地址解析协议,作用是根据目标设备的MAC地址，查询到目标设备的IP地址

```

```
在OSI模型中，第三层网络层负责 IP地址，第二层数据链路层则负责 MAC地址。因此一个主机会有一个MAC地址，而每个网络位置会有一个专属于它的IP地址。在不同的网络状态环境中，你的IP地址是有可能会发生改变的。
```

```
网络操作命令：
ipconfig    查看本地的网络配置信息
ipconfig/all    详细查看本地的网络配置信息

ping  检查你的网络通不通
ping  ip地址    检测当前的主机是否和ip地址所对应的主机能否通信
ping 192.168.0.32

ping  域名(网站的名字)
ping www.baidu.com

查看端口号：netstat  -a  -n
```

TCP/IP网络互联层(重点)

```
功能
在不同网络之间进行路由寻址、传递数据报
IP( Internet Protocol)协议
无连接、不可靠的协议

ICMP协议:因特网控制消息协议
是ip协议的一部分
作用是报告错误,典型应用:ping命令的执行就是icmp协议工作的过程

```

TCP/IP传输层(重点)

```
作用:建立应用间端(发送端)到端(接收端)的连接
    面向连接：会话建立，数据传输，会话拆除(建立维护拆除)  可靠
    无连接：不保证数据的有序到达，不可靠
Tcp协议:也叫传输控制协议 (浏览器)
    面向连接
    可靠（三次握手,四次挥手）
    速度慢
UDP协议:用户数据报协议 (QQ,WX)
    无连接
    不可靠
    速度快

端口号：
我们用户在识别或者认识一个软件，是根据软件的名字识别，
而计算机系统或者网络，是通过端口号来识别软件
```

TCP/IP应用层(重点)

主要负责用户和应用程序之间的通信。协调设备和软件的多样性问题；解决系统中文件传输问题。以下是常见的应用协议：
FTP：文件传输协议 （专门文件传输的协议）
HTTP：超文本传输协议 （网页）
DNS：域名系统
Telnet：远程终端协议 （远程操作需要的协议）
IMAP：Internet邮件访问协议 （针对邮箱,会删除邮件）
POP3：邮局协议版本3 （针对邮箱,不会删除邮件）
SMTP协议又叫:简单邮件传输协议

DNS域名系统

1.因为每个服务器都有一个ip地址,这个ip地址就是服务器的门牌,但是,ip地址不太好记,所以人们发明了DNS域名系统
2.一个ip地址对应一个有意义的字符串,这个字符串分为3段
www.baidu.com
3.当你访问www.baidu.com时,是从本地的DNS系统中查找这个域名对应的ip地址,然后再去访问百度的服务器
如果本地没有,那么就逐层往上的计算机找,如果找到,先存储到本地的域名系统,方便下次使用,然后在转换为ip地址

TCP 传输控制协议

TCP是面向连接的，可靠的，基于流的传输层协议。

TCP报文格式

偏移	字节	0								1								2								3							
字节	比特	0	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	29	30	31
0	0	来源连接端口																目的连接端口														
4	32	序列号码																															
8	64	确认号码（当ACK设置）																															
12	96	数据偏移				保留 0 0 0			NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	窗口大小															
16	128	校验和																紧急指针（当URG设置）														
20 ...	160 ...	选项（如果数据偏移 > 5，需要在结尾添加0。） ...																															

面试题:TCP连接的三次握手

客户端随机生成序列号，发送SYN给服务器端；

```
c6 05 1f 48 e6 a8 6c 68 00 00 00 00 80 02 fa f0
b6 ce 00 00 02 04 05 b4 01 03 03 08 01 01 04 02

源端口: c6 05
目的端口： 1f 48
序列号： e6 a8 6c 68
确认号： 00 00 00 00
数据偏移，保留，NS:  80  02
```

服务器端随机生成序列号，将客户端的序列号加1作为确认号， 发送SYN/ACK给客户端；

```
1f 48 c6 05 c4 a7 ff 74 e6 a8 6c 69 80 12 72 10
41 b3 00 00 02 04 05 78 01 01 04 02 01 03 03 07

源端口: 1f 48
目的端口： c6 05
序列号：c4 a7 ff 74
确认号：e6 a8 6c 69
数据偏移，保留，NS: 80 12
```

客户端将服务器端的序列号加1作为确认号，发送ACK给服务器端。

面试题:TCP建立连接为什么要三次握手，两次可不可以？

```
TCP是可靠的传输层协议，其可靠性是通过数据报文中的序列号来保障的。所以在建立TCP连接的过程当中主要就是为了同步序列号，而数据传输是双向的，对于双向传输的序列号都需要一个确认同步的过程，至少得经过三次数据的交互，所以需要三次握手，两次不可以
```

面试题:TCP断开连接的过程 （4次挥手）

1. ```
   断开TCP连接可以是由客户端发起，也可以是由服务器端发起
   假设断开连接是由客户端发起的：
   1.客户端向服务端发送FIN
   2.服务端向客户端发送ACK   到这,客户端明确不能接收和发送数据给服务端
   3.服务端向客户端发送FIN
   4.客户端向服务端发送ACK 到这,服务端明确不能接收和发送数据给客户端

   ```

面试题:TCP断开连接的过程为什么要4次

```
TCP协议是双工的，断开连接的时候发送的FIN只是表示单向的数据已经传输完毕了，不需要再发送数据，但是可以接收对方发过来的数据。所以要将双向通信完全关闭，需要分别发送一次FIN和返回一次ACK
```

udp
```

```
User Datagram Protocol 用户报文协议

UDP是无连接的，不可靠的，基于用户报文的传输层协议
```

TCP和UDP的区别 (重点)

```
TCP是面向连接的，可靠的，基于流的传输层协议
UDP是无连接的，不可靠的，基于用户报文的传输层协议
因为TCP需要建立连接和断开连接，所以TCP的速度比UDP慢
因为TCP需要维护系列号和确认号等控制信息，所以TCP的报文长度比UDP大
```