多敲多敲多敲！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！！

网络编程

网络功能：数据传输。

ISO （国际标准化组织）

OSI七层模型🡪 网络通信工作流程的标准化

应用层：提供用户服务，具体功能由特定的程序而定

表示层：数据的压缩优化，加密

会话层：建立应用级的连接，选择传输服务

传输层：提供不同的传输服务。流量控制

网络层：路由选择，网络互连

链路层：进行数据转换，具体消息的发送，链路连接

物理层：物理硬件，接口设定，网卡路由交换机等

Cookie

高内聚：模块功能尽可能单一，不要掺杂

低耦合：模块之间尽可能减少关联和影响

OSI七层模型的优点 ：

1. 将工作流程标准化
2. 降低了模块间的耦合度，使每一部分可以单独开发，单独工作。

四层模型

应用层：应用层，表示层，会话层

传输层

网络层

物理链路层

五层模型（TCP/IP模型）

应用层：应用层， 表示层， 会话层

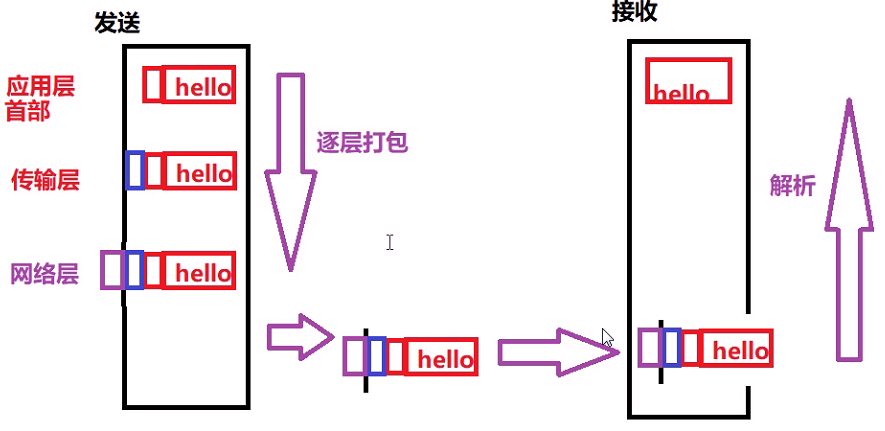
传输层

网络层

链路层

物理层

数据传输流程：



1. 发送端由应用层到物理层逐层添加信息头（首部），最终在物理层发送。
2. 中间经过节点（交换机，路由器等）转发，发送到接收端
3. 在接收端根据发送端的每个信息进行解析，最终消息到应用层展示给用户。

网络协议：在网络通信中双方都遵循的规定。包括建立什么样的网络结构，消息结构，表示代表什么等。

应用层：TFTP HTTP DNS SMTP

传输层：TCP UDP

网络层：IP

物理层：IEEE

网络相关概念

网络主机：在网络中标识一台计算机 HOST

本地使用：’localhost’

‘127.0.0.1’表示测试IP

网络使用：’0.0.0.0’ 自动选择当前计算机的可用网卡地址

查询IP ：Linux：ifconfig 看IP信息

ipython3

In [1]: import socket

获取计算机名

In [2]: socket.gethostname()

Out[2]: 'tedu'

通过计算机名获取地址

In [4]: socket.gethostbyname('tedu')

Out[4]: '127.0.1.1'

IP地址

网络上确定一台主机网络位置的地址

ipv4 ：点分十进制 192.168.1.2 0~255 32位

ipv6：128位

pingip:测试和某台网络 主机可否联通

特殊IP地址

127.0.0.1 本地测试IP

0.0.0.0 使用本机可用IP

192.168.1.0 使用网段IP

192.168.1.1 网关IP

192.168.1.255 广播地址

通过地址获取主机网络信息

In [4]: import socket

In [5]: socket.gethostbyaddr('localhost')

Out[5]: ('localhost', [], ['127.0.0.1'])

用户名 别名 网络地址

点分十进制转换成二进制

In [6]: socket.inet\_aton('192.168.1.2')

Out[6]: b'\xc0\xa8\x01\x02'

二进制地址转换为点分十进制

In [8]: socket.inet\_ntoa(b'\xc0\xa8\x01\x02')

Out[8]: '192.168.1.2'

域名：网络服务器地址的名称

1. 方便记忆
2. 名称表达一定的含义

网络端口号

端口是网络地址的 一部分，用于区分一个网络主机上的网络应用

\*在一个操作系统中不同的网络应用监听不同的端口号

取值范围：1—65535

1-255 一些众所周知的通用端口

256-1023 系统应用端口

1024---65535 自用端口

建议使用 >10000

获取一个应用的端口信息

In [9]: socket.getservbyname('mysql')

Out[9]: 3306

网络字节序：数据在网络中的传输格式

传输层服务

面向连接的传输服务

给予TCP协议的数据传输

传输特征：提供可靠的数据传输，可靠性指数据传输过程中 无丢失，无失序，无差错，无重复。

实现手段 ：数据传输断开前都需要进行传输和断开的确认

三次握手：TCP传输在数据传输前建立连接的过程

1. 客户端向服务端发送连接请求
2. 服务端收到请求后，回复确认消息，表示允许连接
3. 客户端收到服务器回复，进行最终标志发送确认连接

四次挥手：TCP传输在连接断开前进行断开确认的过程。

1. 主动方发送报文告知被动方要断开连接
2. 别动收方到请求后立即返回报文告知已经准备断开
3. 别动方准备就绪后再次发送报文告知可以断开
4. 主动方发送消息，确认最终断开。

应用情况：适用于传输较大的文件，网络情况良好，需要保证传输可靠性的情况。

比如：网页的获取，文件下载，邮件传输，登录注册

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

面向无连接的传输服务

基于udp协议的传输

传输特点：不保证传输的可靠性，传输过程没有连接和断开的流程，数据收发自由。

使用情况 ：网络环境较差，对传输可靠性要求 不高，需要提升传输效率。不变连接，需要灵活收发消息。

比如：网络视频，群聊，广播发送。

要求：

1. OSI七层模型介绍一下，TCP、ip 模型呢

2、TCP服务和udp服务有什么区别

1. 三次握手 和四次挥手 是什么意思，过程是怎么样的

socket套接字编程

目标：根据socket模块提供的接口函数，进行组合使用完成基于TCP或者udp的网络编程。

套接字：完成上述目标的一种编程手段，或编程方案。///网络编程的一种方式

套接字的分类：

流式套接字（sock\_stream）：传输层基于TCP协议的套接字编程方案。

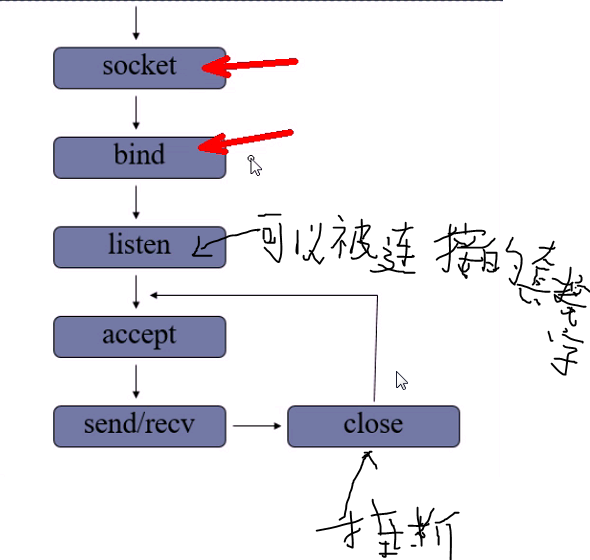
数据报套接字（sock\_dgram）：传输层基于udp协议的套接字编程方案。

底层套接字（sock\_ram）：访问底层协议的套接字编程。

\*面向连接的传输--TCP协议--可靠地--流式套接字

\*面向无连接的传输--udp协议---不可靠----数据报套接字

TCP套接字服务端编程



关闭

消息收发

处理客户端请求

监听套接字

绑定套接字

创建套接字

创建套接字

绑定

监听

处理客户端的连接请求

消息的收发

关闭套接字

*Import socke*t

1. 创建套接字

Sockfd=socket.socket(socket\_family = AF\_INET,

Socket\_type = SOCK\_STREAM,

Proto =0)

功能：创建套接字

参数：socket\_family : 选择地址族类型 AF\_INET

Socket\_type ：套接字类型 SOCK\_STRAM流式

SOCK\_DGRAM数据报

Proto : 选择子协议类型

返回值：返回套接字对象

1. 绑定服务端地址

Sockfd.bind(addr)

功能：绑定IP地址

参数：元组（IP，port）

| |

ip 端口

localhost 可以被本机用127.0.0.1

127.0.0.1 同上

192.168.205.127 可以被所有人用192.168.205.127访问

0.0.0.0 可以被别人用192.168.205.127访问，也可以别自己用127.0.0.1访问。

1. 设置监听套接字

Sockfd.listen(n)

功能：将套接字设置为套接字，创建监听队列

参数：n表示监听队列大小

\*一个监听套接字可以可以连接多个客户端套接字

1. 等待处理客户端连接请求

Connfd,addr=sockfd.accept()

功能：阻塞等待处理客户端连接

返回值：connfd 客户端连接套接字

Addr 连接的客户端地址

\*阻塞函数：程序运行过程中遇到阻塞函数则暂停运行，直到某种阻塞条件达成在继续运行。

5、消息收发

正整数

Data = Connfd.recv(buffersize)

功能：接收对应客户端消息

参数：依次最多接收多少字节

返回值：接收到的内容

\*如果没有消息则会阻塞。

N=connfd.send(data)

功能：大宋消息对应客户端

参数：要发送的内容，必须是bytes格式

返回值：返回实际发送消息的大小

6、关闭套接字

Sockfd.close()

功能：关闭套接字

TCP客户端

1. 创建套接字

\*必须相同类型的套接字才能通信

2、建立连接

Sockfd.connect(servr\_addr)

功能： 建立连接

参数：元组，服务端地址

3、消息收发

\*消息收发要和服务端配合，避免两边都出现recv阻塞

4、关闭套接字

