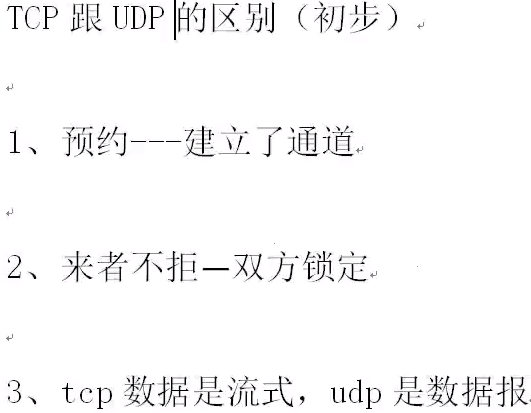
学习资料：《[自制 windows网络编程](https://www.bilibili.com/video/av3906340/)》-- 书籍：《windows\_socket》 群：186953317

1. tcp和udp



1. udp情况下，recvfrom可以判断对方无响应（关机等），只要我们先发送过数据就行。如果对方无响应，recvfrom 返回 0。
2. 转换函数和获取网络地址函数

inet\_addr 字符串IP转换成网络字节序的数字 -> 变数字 后 转网络

htonl 主机字节序 转换成 网络字节序 ulong -- 认为是用在转换数字IP

htons 主机字节序 转换成 网络字节序 short -- 认为是用在转换端口

逆反：网络字节序变成主机字节序

ntohl/ntohs -- 数字变数字

inet\_aton（逆inet\_addr） -- 网络字节序数字IP转变为主机字节序字符串IP

获取网络地址信息:需要用代码获取，

通过域名获取IP -> gethostbyname,返回一个结构体，结构体是对方主机信息，包括IP地址；

获取本地主机名 -> gethostname

获取本地IP -> 没有API，需要分两步进行 -> 获取本地主机名，在根据主机名获取IP

通过IP获取主机信息-> gethostbyaddr

我们需要注意gethostname可能返回空的问题

异步模式WSAgethostbyname

1. TCP/UDP/流式/数据包

4.1 tcp只能流式，udp只能数据包

流式特性 -- 粘包：他只保证顺序，但是中间可以任意粘在一起，也可能把本来一块的给分开。Tcp流式可以任意的切割，一次send如果太大，对方可以多次recv，但是udp数据包不行。

Udp数据是独立的，不可分割的，一次send 就对应一次 recv，不可能粘包，容纳不了就丢弃。

4.2 tcp有自动校验和重发机制，udp有校验但没有重发，只能丢弃。

4.3 tcp用于大数据量传输(网盘下载，迅雷)，UDP用于小量/广播 不可靠消息

4.4 tcp 可人为设置断点续传，udp一般不会去设置

五、recv send异常原因和应对

1、recv异常

1.1、对方send异常

1.1.1 对方程序或机器故障或关机

1.1.2 数据丢失

1.2我方recv异常

1.2.1 发现数据长度不符合

1.2.2 UDP超过长度导致丢包

1.3网络中途异常

1.3.1 网络中途数据丢失

1. send异常

2.1 我方send异常

2.1.1 本地发送时就丢失数据（根据返回值） --- 异步非阻塞特例

2.2 对方recv异常

2.2.1 对方程序或机器故障或关机

2.2.2 UDP下对方缓冲区不够导致丢包

2.3 网络中途异常

2.3.1 网络中途数据丢失

六、非阻塞模型 -- 六模型之一 <https://www.bilibili.com/video/av3906340/?p=8>

1. 非阻塞模型(把套接字设置成非阻塞)
   1. 原理

把套接字设置为非阻塞性质（查看有没有+sleep+循环），那么之后的accept 和 recvfrom 都会立即返回。因为立刻返回，所有accept 和 recvfrom 都是写在死循环里面(也就是个人让他隔一段时间去看看<靠sleep 函数>，有就立即接受返回，没有就返回)。

* 1. 优点：配合死循环和sleep，在sleep的时候会可以干其他事情
  2. 缺点：不断频繁的去调用(查看)，造成CPU的较大浪费。如果sleep比较长，那么可能产生滞后，对于响应要求快的则不适合。
  3. 代码
     1. 核心设置套接字非阻塞:[ioctlsocket](https://blog.csdn.net/kaibing/article/details/78199849)

// 设置为非阻塞模式

Int imode = 1;

retVal=ioctsocket(sServer, FIONBIO,(u\_long\*)&imode);

1. select模型
2. 功能概述：通过一些步骤去监视套接字（选入套接字），调用select函数来得知有没有信息，有信息则处理，处理完以后就进入循环调用select。

FD\_SET(socket, &结构体)

通过以上，把指定套接字加入结构体中，用这个结构体来管理多个套接字（再添加就再调用，一次添加一个）。

然后循环调用select函数，就能监测结构体里多个套接字是否有信息。只要其中一个有，就返回提示。

然后用户自己去遍历判断。

2、说明：

2.1select 可以选择你要监听的类型（有可读:比如 recvfrom，注意listen后可以accept似乎也是可读）。所以，我们选入的套接字，可以是监听套接字，也可以是后面收发数据的套接字，往往循环里会先选入监听，后面检测到后accept，再选入收发套接字。

2.2似乎select还能指定监测的时间间隔，不过刘炎windows网络编程的举例里，没有设置时间间隔。

3、意义

3.1 其实，感觉就是提供了一个遍历多个套接字的功能。这样子我们再一个线程里面，也可以完成并发的效果（多个套接字再工作）

3.2 他是监测（提取要我们自己来），而非阻塞是直接去提取。

3.3 监测的内容很广，几乎涵盖了所有的信息

4、select优缺点

4.1 优点

4.1.1 select用单线程实现并发效果(因为能操作多个套接字)，如果是阻塞模式的并发，那就要开辟多个子线程，所以select在一定程度上避免了线程膨胀。

4.1.2 select函数，他的超时设置功能，可以应用在高精度实时性的环境中，他的时间精度是微秒。

4.2 缺点

4.2.1 管理的套接字有限 只能64个

4.2.2 大量频繁调用select增加开销