**题目：TCP协议设计与实现**

**姓名: 王翰墨**

**学号：2019302683**

**班号：10011902**

**课程代号：网络试点班**

**计算机学院**

**时间:2022年1月1日**

**目 录**

摘 要

[1 目的 1](#_Toc341302685)

[2 要求 1](#_Toc341302686)

[3 相关知识 1](#_Toc341302687)

[4 实现原理 4](#_Toc341302688)

[5 运行结果与分析 6](#_Toc341302690)

[6 参考文献 7](#_Toc341302690)

# 题目：TCP协议设计与实现

# 目的

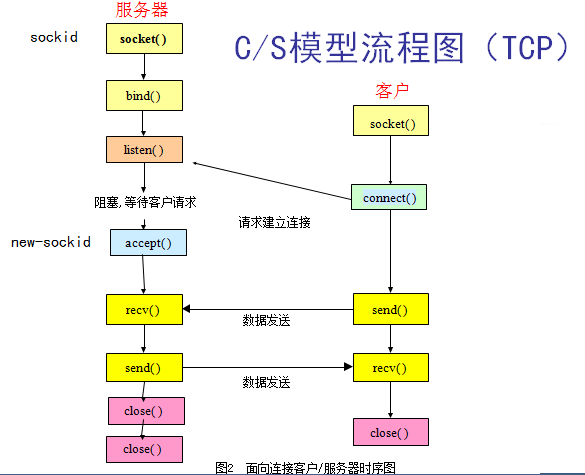
在网络协议栈中对TCP协议的支持，并利用TCP协议传输数据。

# 2、要求

实现TCP的socket(),bind(),connect(),listen(),accept(), send(),recv(),close()函数功能，使其能支持客户端/服务器模型的通信接口。在TCP协议通信前进行三次握手，双方协商MSS和窗口大小，随后进行通信。通信过程中保证可靠性，接收到数据后发送ACK应答，通信结束时进行四次挥手。

# 3、相关知识

TCP协议通信模型



客户端和服务器定义所有全局变量为一个数据数据结构，数据结构对应变量为动态分配。

该数据结构内容包括：

客户端初始序号：

服务器初始序号：

发送窗口大小：字节为单位

发送缓存大小：字节为单位

接收缓存大小：字节为单位

接收窗口大小；字节为单位

客户端MSS大小；

服务器MSS大小；

通信MSS大小；

（1）创建套接字old\_socket，其调用格式如下：

old\_sockid＝socket（af，type，protocol）

功能：

构造好一个通信五元组数据结构变量，并完成初始化；

（2）bind( ) - 用于TCP服务器端old\_sockid对象上绑定进程标识（服务器IP地址+周知端口号），bind( )的调用格式为：

bind（old\_sockid，Servaddr，addrlen ）

功能：

绑定服务器ＩＰ地址和端口号

1. 建立连接请求connect( ) :客户端调用

采用TCP协议通信的客户通过调用connect( ) 主动请求与服务器建立连接；connect() 的调用格式为：

connect（sockid，Seraddr，Seradd-len ）

(4)服务器端listen( )：服务器端

采用TCP协议的服务器进程一般在某个周知的端口上等待客户进程的连接请求（客户-connect()，服务器-*listen( )*）。

Listen（）函数调用规范如下：

Listen（old\_sockid，quelen）

（5）accept( )：服务器进程处理客户进程的连接请求；

调用方法：new-sockid = accept（old\_sockid，clientaddr，addrlen）

(6) TCP发送函数：send（sockid，buf，buflen，flags）

采用TCP协议进行数据发送，任意一方可以向对方发送数据；可以不必指定接收方socket地址，因为sockid中已包含通信的5元组:

send（sockid，buf，buflen，flags）

（7）TCP协议数据接收函数

recv（sockid，buf，buflen，flags）

（8）释放所占有的资源，服务器端和客户端使用

int closesocket( SOCKET socketid ) // socketid为欲关闭的套接字；

用法：  
 int nResult=closesocket(socketid);

# 4、实现原理

**总体构思：**如下图所示，不改变Ethernet和IPv4协议的四线程生产者-消费者处理流程，在网络层交付时，根据上层协议字段进入TCP接收队列或者UDP接收队列，传输层交付时，统一送入IPv4发送队列，但需要标记一下自身是TCP还是UDP协议。该网络协议栈支持Ethernet、IPv4、ARP、ICMP、UDP和TCP。

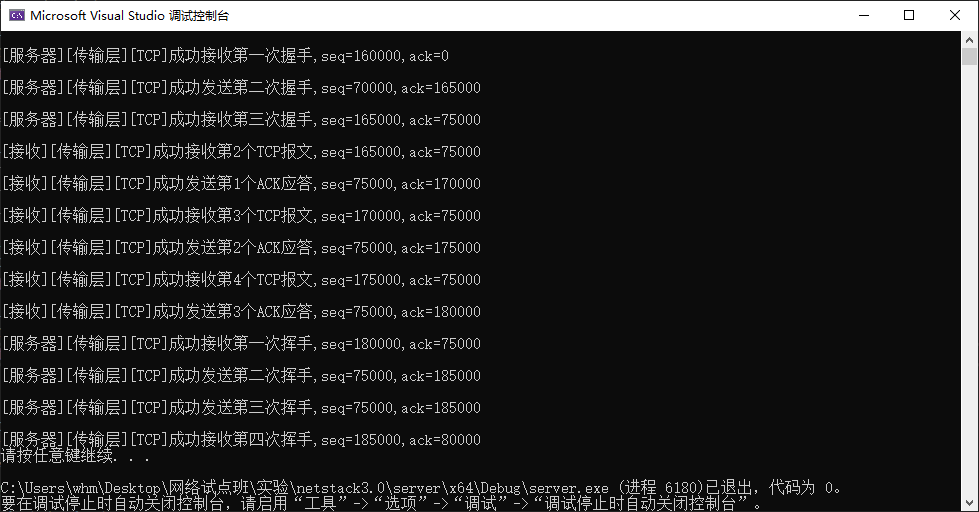


**实现TCP的思路：**在服务器端首先通过socket()创建一个旧套接字，用bind()绑定服务器的IP地址和端口号，随后调用listen()侦听TCP链接请求。在TCP接收队列中找到一个SYN位有效的报文段，则将其保存下来，交给accept()函数处理。accept()中根据客户端的IP地址和端口创建新套接字，并且发送第二次握手的TCP报文（SYN和ACK有效），并接收第三次握手。成功握手后，服务器调用recv()持续接收客户端发送过来的TCP报文，每成功收到一个TCP报文就构造一个相应的ACK应答并且发送给客户端。如果接收到了FIN位有效的报文段（第一次挥手），则发送ACK应答（第二次挥手）并停止接收，转而准备进入close()函数。在close()中服务器进行第三次和第四次挥手，并且释放掉动态分配的全局变量，将套接字还原回旧套接字。

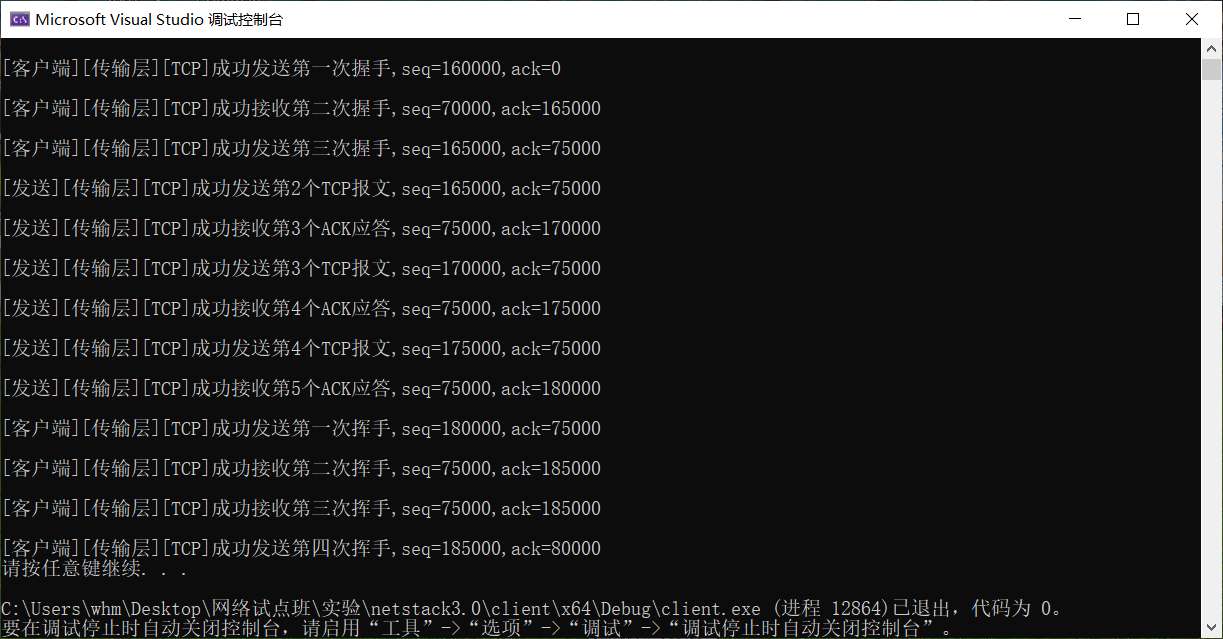
在客户端，首先通过socket()创建套接字，接着调用connect()函数向服务器端发送TCP链接请求，即发送第一次握手，接收第二次握手，再发送第三次握手。连接建立后，客户端调用send()将需要发送的数据按MSS划分成多个字节流，并封装成多个TCP报文段交付给网络层。发送完成后调用close()函数发送第一次挥手，接收第二次挥手，然后接收第三次挥手并发送第四次挥手，成功释放连接后，将动态分配的套接字和全局变量释放，结束通信。

# 5、运行结果与分析

**服务器：**在连接阶段，服务器接收第一次握手、发送第二次握手、接收第三次握手。在数据传输阶段，服务器接收到了三个TCP报文段，并且发送对应的ACK应答。在释放阶段，服务器接收第一次挥手、发送第二次挥手、发送第三次挥手、接收第四次挥手。



**客户端：**在连接阶段，客户端发送第一次握手、接收第二次握手、发送第三次握手。在数据传输阶段，客户端发送了三个TCP报文段，并且接收到了每个对应的ACK应答。在释放阶段，客户端发送第一次挥手、接收第二次挥手、接收第三次挥手、发送第四次挥手。



# 6、参考文献

* 《计算机网络协议分析与实践》姚烨,朱怡安 电子工业出版社