**中国矿业大学计算机学院**

**21 级本科生 计算机网络实验 报告**

实验内容 基于TCP的网络应用编程

学生姓名 杨学通 学 号 08213129

专业班级 人工智能一班

任课教师 徐秀老师

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **课程基础理论掌握程度** | 熟练 🞏 | 较熟练 🞏 | 一般 🞏 | 不熟练 🞏 |
| **综合知识应用能力** | 强 🞏 | 较强 🞏 | 一般 🞏 | 差 🞏 |
| **报告内容** | 完整 🞏 | 较完整 🞏 | 一般 🞏 | 不完整 🞏 |
| **报告格式** | 规范 🞏 | 较规范 🞏 | 一般 🞏 | 不规范 🞏 |
| **实验完成状况** | 好 🞏 | 较好 🞏 | 一般 🞏 | 差 🞏 |
| **工作量** | 饱满 🞏 | 适中 🞏 | 一般 🞏 | 欠缺 🞏 |
| **学习、工作态度** | 好 🞏 | 较好 🞏 | 一般 🞏 | 差 🞏 |
| **抄袭现象** | 无 🞏 | 有 🞏 姓名: | | |
| **存在问题** |  | | | |
| **总体评价** |  | | | |

综合成绩： 任课教师签字：

年 月 日

**中国矿业大学计算机学院实验报告**

|  |
| --- |
| 课程名称：计算机网络实验 实验名称：基于TCP的网络应用编程  班级：人工智能一班 姓名：杨学通 学号：0813129 实验日期：2023.6.19  实验报告要求：1.实验目的2.实验内容3.实验步骤，4.运行结果5.流程图6.实验体会 |
| 实验一：基于TCP的socket应用程序设计  一 实验要求  分别设计基于TCP的client和server程序，并在两个不同的主机上运行，实现文件的上传与下载功能；在两台主机上分别运行抓包软件，观察建立连接、传输数据与释放连接的过程  二 实验目的  熟悉基于TCP的C/S网络通信过程，掌握Socket的数据结构及常用的Socket函数，用C/C++语言编写一个完整的C/S模式的TCP网络应用程序并进行相应的运行、测试与抓包分析。  三 实验步骤  5.1.1 编写Server.cpp代码  #include <Winsock2.h>  #include <stdio.h>  #pragma comment(lib,"ws2\_32.lib")  int main(){  WORD wVersionRequested; // 指定准备加载的Winsock库版本  WSADATA wsaData; // Winsock库版本信息的结构体  wVersionRequested = MAKEWORD( 1, 1);  int err = WSAStartup( wVersionRequested, &wsaData ); // 加载套接字库  if ( err != 0 ) { return 0;}  if ( LOBYTE( wsaData.wVersion ) != 1 ||HIBYTE( wsaData.wVersion ) != 1 ) {  WSACleanup( );// 释放为该应用程序分配的资源，终止对WinSock动态库的使用  return 0;  }  SOCKET sockSrv=socket(AF\_INET,SOCK\_STREAM,IPPROTO\_TCP);  // 创建套接字AF\_INET表示TCP/IP协议  // SOCK\_STREAM表示TCP连接，SOCK\_DGRAM表示UDP连接  // IPPROTO\_TCP表示TCP协议  SOCKADDR\_IN addrSrv; // 定义一个地址结构体的变量  addrSrv.sin\_addr.S\_un.S\_addr=inet\_addr("192.168.220.199");  addrSrv.sin\_family=AF\_INET;  addrSrv.sin\_port=htons(6000);//htons把一个u\_short类型从主机字节序转换为网络字节序  bind(sockSrv,(SOCKADDR\*)&addrSrv,sizeof(SOCKADDR)); //将套接字绑定到本地的某个地址和端口上  listen(sockSrv,5); //将指定的套接字设定为监听模式  SOCKADDR\_IN addrClient;  int len=sizeof(SOCKADDR);  while(1) {  SOCKET sockConn=accept(sockSrv,(SOCKADDR\*)&addrClient,&len);  //接受客户端发送的连接请求  char sendBuf[100];  sprintf(sendBuf,"Welcome %s to here",inet\_ntoa(addrClient.sin\_addr));  send(sockConn,sendBuf,strlen(sendBuf)+1,0);  //通过一个已建立连接的套接字发送数据  char recvBuf[100];  recv(sockConn,recvBuf,100,0); //从一个已建立连接的套接字接收数据  printf("%s\n",recvBuf);  closesocket(sockConn);  }  return 0;  }  5.1.2 编写Client.cpp代码  #include <Winsock2.h>  #include <stdio.h>  #pragma comment(lib,"ws2\_32.lib")  int main(){  WORD wVersionRequested; // 指定准备加载的Winsock库版本  WSADATA wsaData; // Winsock库版本信息的结构体  wVersionRequested = MAKEWORD( 1, 1);  int err = WSAStartup( wVersionRequested, &wsaData ); // 加载套接字库  if ( err != 0 ) { return 0;}  if ( LOBYTE( wsaData.wVersion ) != 1 ||HIBYTE( wsaData.wVersion ) != 1 ) {  WSACleanup( );  // 释放为该应用程序分配的资源，终止对WinSock动态库的使用  return 0;  }  SOCKET sockClient=socket(AF\_INET,SOCK\_STREAM,IPPROTO\_TCP);  // 创建套接字。AF\_INET表示TCP/IP协议  // SOCK\_STREAM表示TCP连接，SOCK\_DGRAM表示UDP连接  //IPPROTO\_TCP表示TCP协议  SOCKADDR\_IN addrSrv; //定义一个地址结构体的变量  addrSrv.sin\_addr.S\_un.S\_addr=inet\_addr("192.168.220.199");  addrSrv.sin\_family = AF\_INET;  addrSrv.sin\_port = htons(6000);  connect(sockClient,(SOCKADDR\*)&addrSrv,sizeof(SOCKADDR));  //向服务器发出连接请求  char recvBuf[1000];  recv(sockClient,recvBuf,100,0); //接收数据  printf("%s\n",recvBuf);  send(sockClient,"Hello,Server!",strlen("Hello,Server!")+1,0); //发送数据  closesocket(sockClient); //关闭套接字  WSACleanup();  system("pause");  return 0;  }  5.1.3 在两台主机上分别运行Server和Client代码    5.1.4 启动Wireshark并抓包    图中，红框表示TCP建立连接时的三报文握手，蓝框表示TCP释放连接时的四次握手。  5.1.5 分析数据包    源IP地址：192.168.220.176  目的IP地址：192.168.220.199  发送方进程端口号:12230  接受请求的信宿进程端口号:6000  字节流序号:4  确认字节流序号:1  头部长度:5 //代表头部长度20字节（5x4）  标志字段:0x2  通知窗口:131328 //告知对方自己的发送能力  校验和:0xc52f  紧急指针:0  与编写的程序完全相符！  四 流程图    **实验体会**  通过本次实验，我学会了用C++编写程序，设计基于TCP的socket的网络通信应用，实现了Client和Server之间相互通信。相比较与UDP协议，TCP协议的通信过程发更加复杂，开发程序具有一定的难度。尤其是TCP连接建立的过程，仍然值得思考。实验后期，通过Wireshark抓取通信过程产生的数据包，分析帧结构，验证了Client和Server程序的正确性。 |