**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UDP的网络应用层编程

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UDP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：2023.6.12  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UDP的socket应用程序设计  一 实验要求  分别设计基于UDP的client和server程序，调试完成后在两个不同主机上运行，测试数据的双向传输功能；分别在两台主机上运行抓包软件，抓取并分析程序运行过程中发送和接收的数据。  二 实验目的  熟悉基于UDP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++语言编写一套完整的C/S模式的UDP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 <iostream>  #pragma comment(lib, "ws2\_32.lib")  int main(void) {  // 初始化套接字库  WORD wVersion;  WSADATA wsaData;  int err;  wVersion = MAKEWORD(1, 1);  err = WSAStartup(wVersion, &wsaData);  if (err != 0) {  return err;  }  if (LOBYTE(wsaData.wVersion) != 1 || HIBYTE(wsaData.wVersion) != 1) {  WSACleanup();  return -1;  }  // 创建套接字  SOCKET sockSrv = socket(AF\_INET, SOCK\_DGRAM, IPPROTO\_UDP);  SOCKADDR\_IN addrSrv;  addrSrv.sin\_addr.S\_un.S\_addr = inet\_addr("192.168.220.176");  addrSrv.sin\_family = AF\_INET;  addrSrv.sin\_port = htons(6001);  // 绑定到本机6001端口  bind(sockSrv, (SOCKADDR \*)&addrSrv, sizeof(SOCKADDR));  // 接收请求，处理请求  SOCKADDR\_IN addrCli;  int len = sizeof(SOCKADDR);  addrCli.sin\_addr.S\_un.S\_addr = inet\_addr("192.167.220.199");  addrCli.sin\_family = AF\_INET;  addrCli.sin\_port = htons(6001);  char sendBuf[] = "Hello，Client ! ";  char recvBuf[100];  std::cout << "start UDP server with port 6001" << std::endl;  while (1) {  // 接收数据  recvfrom(sockSrv, recvBuf, 100, 0, (SOCKADDR \*)&addrCli, &len);  std::cout << "Recv:" << recvBuf << std::endl;    // 发送数据  sendto(sockSrv, sendBuf, strlen(sendBuf) + 1, 0, (SOCKADDR \*)&addrCli, len);  std::cout << "Send:" << sendBuf << std::endl;  }  closesocket(sockSrv);  WSACleanup();  return 0;  }  5.1.2 编写Client.cpp代码  #include <WinSock2.h>  #include <iostream>  #pragma comment(lib, "ws2\_32.lib")  int main(void) {  WORD wVersion;  WSADATA wsaData;  int err;  wVersion = MAKEWORD(1, 1);  err = WSAStartup(wVersion, &wsaData);  if (err != 0) {  return err;  }    if (LOBYTE(wsaData.wVersion) != 1 || HIBYTE(wsaData.wVersion) != 1) {  WSACleanup();  return -1;  }  SOCKET sockCli = socket(AF\_INET, SOCK\_DGRAM, IPPROTO\_UDP);  SOCKADDR\_IN addrSrv;  addrSrv.sin\_addr.S\_un.S\_addr = inet\_addr("192.168.220.176");  addrSrv.sin\_family = AF\_INET;  addrSrv.sin\_port = htons(6001);  //服务器套接字  SOCKADDR\_IN addrCli;  addrCli.sin\_addr.S\_un.S\_addr = inet\_addr("192.168.220.199");  addrCli.sin\_family = AF\_INET;  addrCli.sin\_port = htons(6001);  //客户端套接字  int len = sizeof(SOCKADDR);  char sendBuf[] = "你好，服务器！"; //发送缓冲区  char recvBuf[100]; //接收缓冲区  std::cout << "send to Server: " << sendBuf << std::endl;  sendto(sockCli, sendBuf, strlen(sendBuf) + 1, 0, (SOCKADDR\*)&addrSrv, len); //向服务器发送缓冲区内容，并打印到屏幕  recvfrom(sockCli, recvBuf, 100, 0, (SOCKADDR\*)&addrSrv, &len);  std::cout << "recv from: " << recvBuf << std::endl;  //接收服务器发来的内容，并打印到屏幕  closesocket(sockCli);  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    5.1.5 分析数据包  源IP地址：192.168.220.199  目的IP地址：192.168.220.176  协议类型：IPv4  源端口（2字节）：e5:46 十进制值58694 DNS客户程序临时端口号  目的端口（2字节）：17:71 十进制值 6001 DNS服务程序知名端口号  数据报长度（2字节）：00 17 十进制值 23  校验和（2字节）：3a f2  负载字节数（15字节=23-8字节）：c4 e3 ba c3 a3 ac b7 fe ce f1 c6 f7 a3 a1 00  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UDP的socket的网络通信应用，实现乐Client和Server之间相互通信。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。 |