# 桂 林 电 子 科 技 大 学

## 计算机网络 实 验 报 告

|  |
| --- |
| 评语：  指导教师签名：  成绩： |

实验名称 实验六LINUX环境下UDP通信程序设计

计算机与信息安全学院 学院

计算机科学与技术 专 业

**姓名 吴河山**

**学号 2000500927**

实 验 日 期2022年 12 月 1 日

**一．实验目的**

1. 熟悉基于socket的网络编程接口
2. 掌握流式套接字的创建方法
3. 掌握为套接字绑定IP地址、端口的方法
4. 加深理解UDP通信双方的交互模式
5. 掌握recvfrom函数用法
6. 掌握sendto函数用法

**二．实验环境**

1. 头歌基于Linux的虚拟机桌面系统
2. 网络报文分析工具：wireshark
3. 编码工具：Vscode（推荐）或 Vim
4. C编译器：gcc
5. 查询Linux C函数用法：man 2 函数名

**三．相关原理或知识点**

**1.UDP协议的主要特点**

（1）无连接通信

（2）不保证可靠性

（3）实时性高于TCP

（4）报文不分段，可以是大报文（有上限），面向报文通信

**2.Socket（套接字）编程接口**

Unix/Linux、Windows等操作系统，为程序员提供了一种基于socket（套接字）的间接访问系统TCP/IP协议栈进行通信的编程接口，目前大多数通信应用程序的编程都直接或间接地使用了该接口。在Windows系统环境这个接口称之为Winsock API接口。

**3、Socket（套接字）编程接口**

Unix/Linux、Windows等操作系统，为程序员提供了一种基于socket（套接字）的间接访问系统TCP/IP协议栈进行通信的编程接口，目前大多数通信应用程序的编程都直接或间接地使用了该接口。在Windows系统环境这个接口称之为Winsock API接口。

**4、创建UDP套接字**

Linux系统提供一个socket系统调用来创建一个套接字。socket函数的具体的说明如下：

需要的头文件如下：#include <sys/types.h> #include <sys/socket.h>

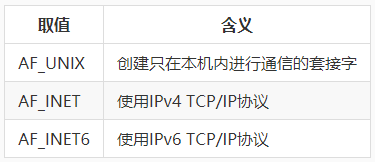
* 函数原型声明： int socket(int domain, int type, int protocol);
* 参数说明：

domain：创建套接字所使用的协议族；

type：套接字类型；

protocol：用于指定某个协议的特定类型，通常某个协议中只有一种特定类型，这样该参数的值仅能设置为0；

* domain参数的常用的协议族如下表所示：



* type参数的常用的套接字类型如下表所示：



函数返回值说明： 执行成功返回值为一个新创建的套接字，否则返回-1，并设置错误代码errno。

**5、IP地址格式转换**

（1）struct in\_addr inet\_addr(char \*) 将一个字符串格式的ip地址转换成一个uint32\_t（32位无符号整数）数值类型IP地址，按网络字节顺序存储。

（2）char\* inet\_ntoa(struct in\_addr) ：把类型为struct in\_addr的数值类型的IP地址转化为字符串格式的ip地址。

**6、绑定端口**

Linux提供了一个bind函数来将一个套接字和某个IP地址、某个端口绑定在一起。bind函数的具体的说明如下：

* 需要的头文件如下：

#include <sys/types.h>

#include <sys/socket.h>

* 函数原型声明如下：

int bind(int sockfd, const struct sockaddr \*addr, socklen\_t addrlen);

* 参数说明：

sockfd：已经创建的套接字；

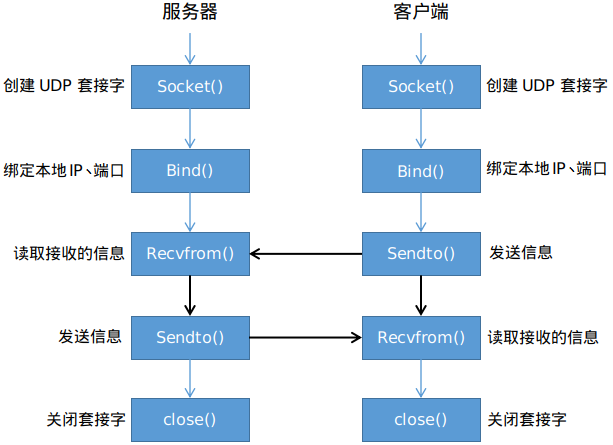
addr：是一个指向sockaddr参数的指针，其中包含了IP地址、端口；

addrlen：addr结构的长度；

* 函数返回值说明：

调用成功，返回值为0，否则返回-1，并设置错误代码errno。

**7、UDP通信两端的交互模式**



**四、实验内容**

1.UDP套接字创建与端口绑定

2.UDP数据发送与接收

3.拓展实验

**五、实验步骤**

**1. UDP套接字创建与端口绑定：**

1）双击打开桌面上的工作区文件夹workspace，再双击实训文件夹myshixun，打开文件message7-1.txt。

2）在/home/headless/Desktop/workspace/myshixun/client/目录下，阅读程序test.c，请用编辑器Vscode，完善/\*\*\*BEGIN\*\*\*/与/\*\*\*END\*\*\*/之间的语句（3处，每处1句）。将这3个语句依次写到文件message7-1.txt的第1、2、3行末尾，要求语句中尽可能不含空格。

3）并保存该文件message7-1.txt。

**2．UDP数据发送与接收：**

1）双击打开桌面上的工作区文件夹workspace，再双击实训文件夹myshixun，打开文件message7-2.txt。

2）在/home/headless/Desktop/workspace/myshixun/client/目录下，阅读客户端程序c2.c。

3）在/home/headless/Desktop/workspace/myshixun/server/目录下，阅读服务器端程序s2.c，请用编辑器Vscode，完善/\*\*\*BEGIN\*\*\*/与/\*\*\*END\*\*\*/之间的语句（3处，每处1句）。将这3个语句依次写到文件message7-2.txt的第4、5、6行末尾，要求语句中尽可能不含空格。

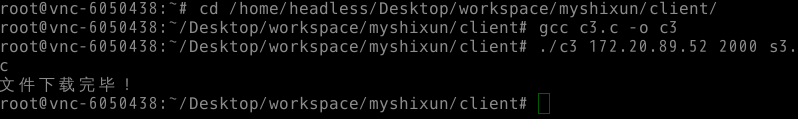
**3. 拓展实验：**

1）参考案例项目设计一个基于UDP的文件下载工具

该工具由服务器端程序s3.c和 客户端程序c3.c两部分组成。要求服务器端先运行，它会提示可下载的文件信息，客户端后运行，它会向服务器端请求下载某个文件，文件不存在会提示，文件下载完毕会提示，然后双方程序结束。服务器端运行时可带一个参数，代表UDP端口，没有参数时端口默认为1234；客户端运行时只能带1个参数或3个参数，带1个参数时，这个参数是要下载的文件名；带3个参数时，它们分别是服务器端IP、服务器端端口、要下载的文件名。

编译运行测试结果如下：





2）双击打开桌面上的工作区文件夹workspace，再双击实训文件夹myshixun， 打开文件message7-3.txt。

3）服务器端程序s3.c已经完成，放在以下目录内：

/home/headless/Desktop/workspace/myshixun/server/

客户端程序c3.c基本已经完成，放在以下目录内：

/home/headless/Desktop/workspace/myshixun/client/

请用编辑器Vscode，完善/\*\*\*BEGIN\*\*\*/与/\*\*\*END\*\*\*/之间的语句（2处，每处1句）。

将这2个语句依次写到文件message7-3.txt的第7、8行末尾，要求语句中尽可能不含空格。

4）分别在两个终端窗口内，编译这两个程序；通过编译后再运行测试这两个程序。

5）把编译后的客户端程序c3，发到自己的邮箱内。在当前的平台上，进入一个终端窗口，用ifconfig命令查看自己主机的IP地址，进入服务器端所在目录，运行s3。

6）在另一个浏览器窗口中，进入上一次实验项目的桌面系统，从自己的邮箱内下载c3到本地（默认目录：/home/headless/下载），在当前的平台上，进入一个终端窗口，用ifconfig命令查看自己主机的IP地址，然后带参数运行c3（3个参数）。

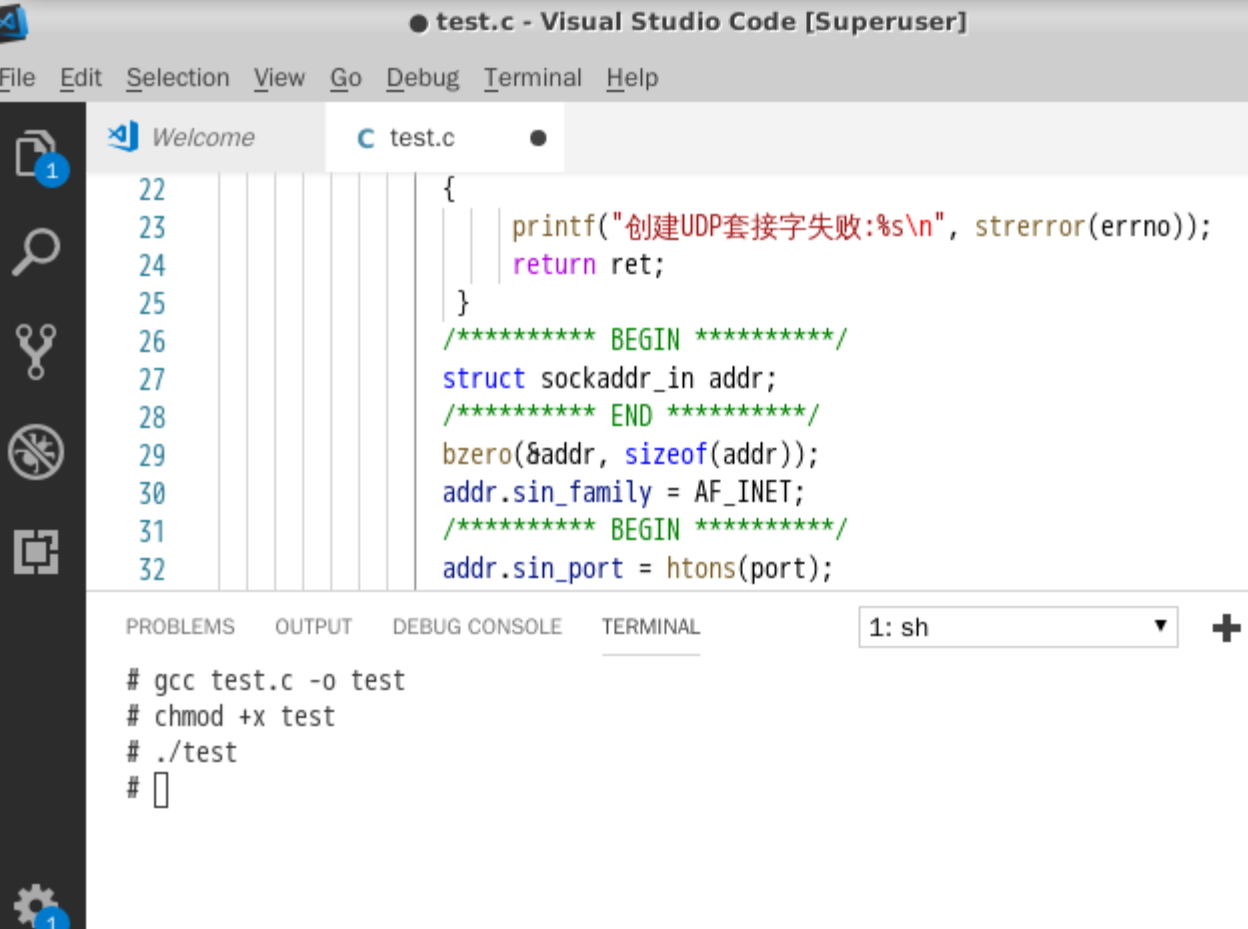
7）针对两端通信报文，要求在服务器端用Wireshark进行跟踪，通信结束后，在Wireshark的显示过滤器栏内，按某一方的IP地址和UDP端口信息进行过滤。

8）分析所捕获的报文，最大的UDP PDU（协议数据单元）长度是多少字节？其在网络层封装时被分成几片？请将答案（只写数值）依次写到文件message7-3.txt的第9行末尾，要求答案之间加空格隔开。

**六、实验结果及其分析**

1. **UDP套接字创建与端口绑定：**

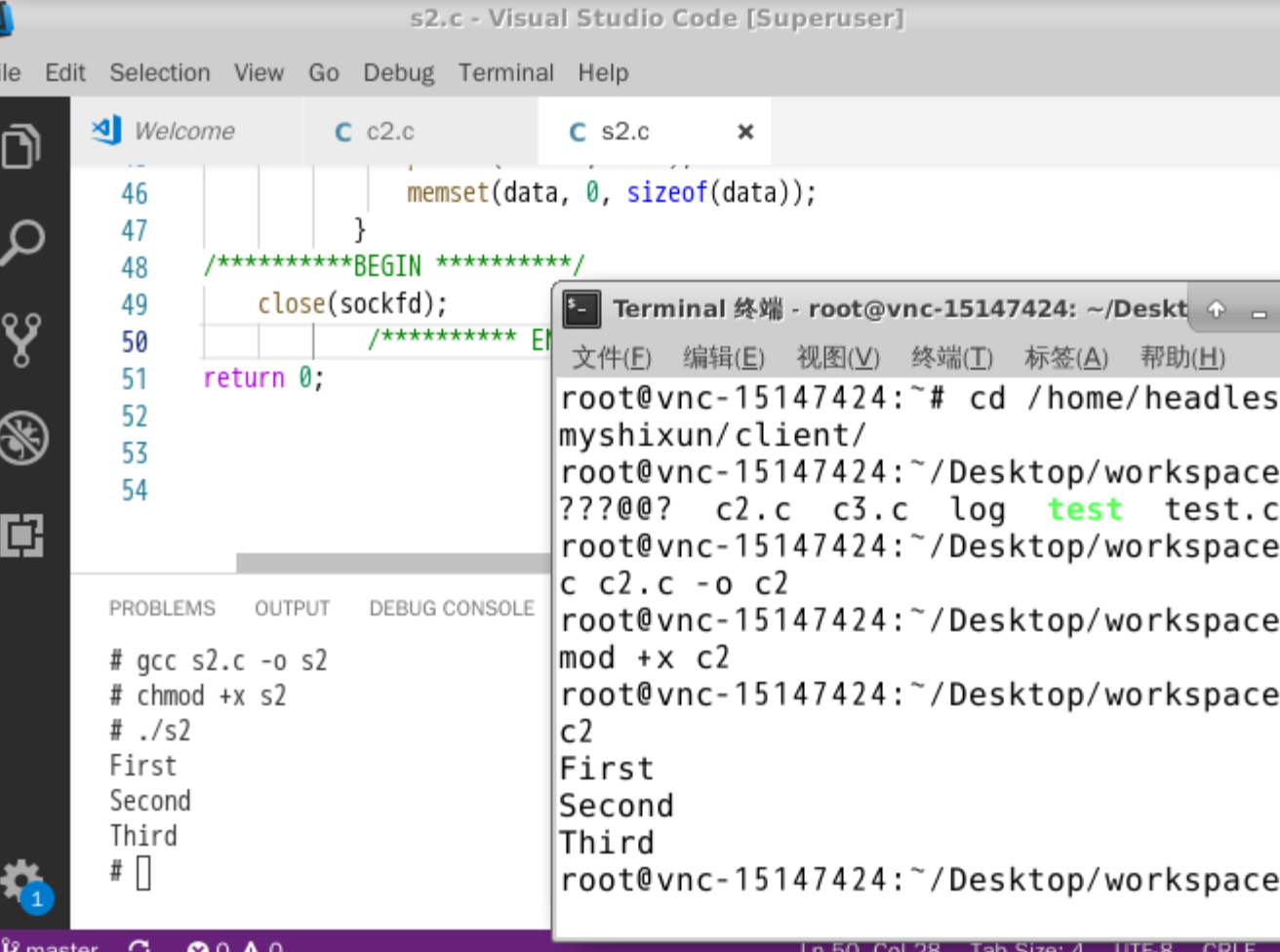




**运行成功。**

**2．UDP数据发送与接收：**

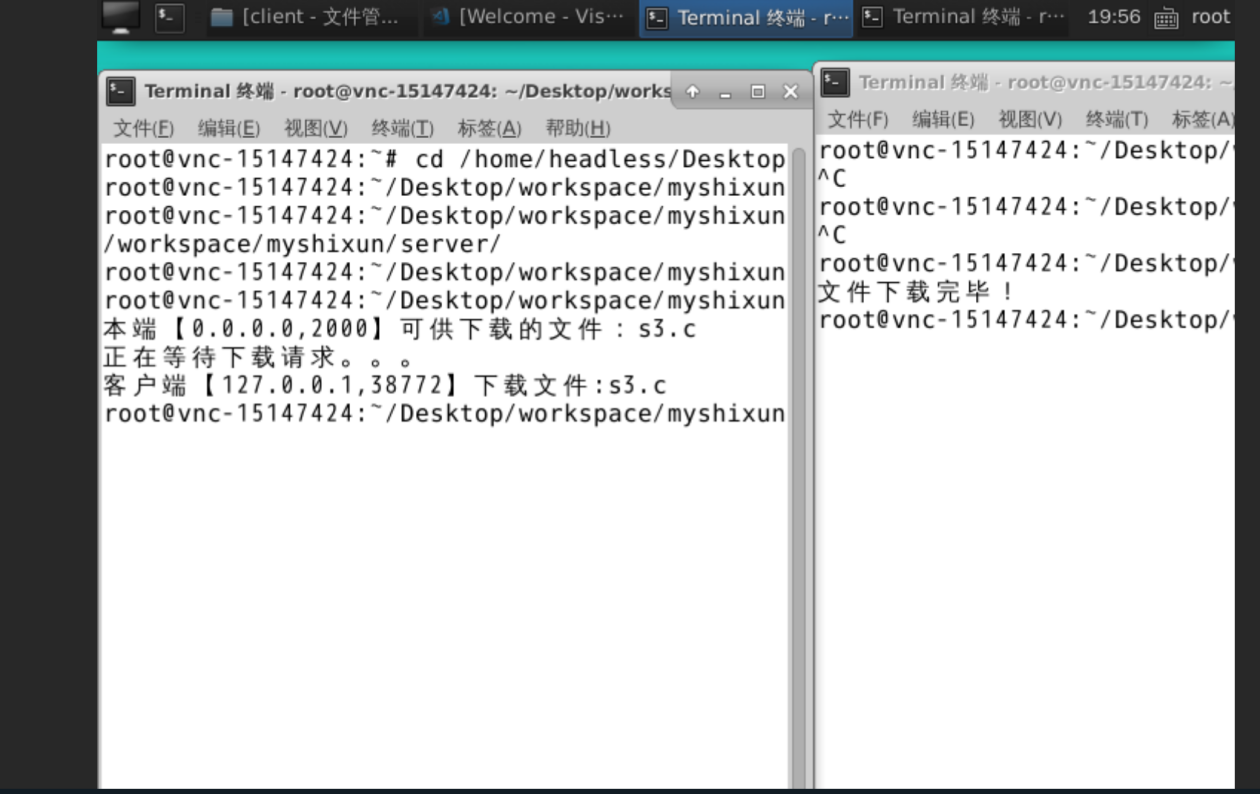




**双方相互交换信息成功。**

**3. 拓展实验：**





**文件下载成功。**

**思考题：**

1、当Socket函数调用成功创建了一个数据报套接字时，系统是否自动地为当前应用分配了一个IP地址和一个UDP端口号？

**不会，需要用bind（）函数进一步将套接字、IP和端口绑定**。

2、在Linux系统下，使用哪条命令可以查看当前系统所有被占用的UDP端口？

**netstat -nulp**

3、设计一对多的通信程序，除了可以采用多线程技术，还可以采用什么技术？

**端口技术，缓冲区重用技术**

**七、实验总结**

通过本次实验，我熟悉了基于socket套接字的编程，学习了套接字的创建，套接字地址的创建，套接字端口的绑定，以及UDP协议数据的接收和发送。通过第三关，建立一个基于UDP的下载工具，了解在编程层面对计算机网络的操作。