

**计算机专业类课程**

**实**

**验**

**报**

**告**

**课程名称： 计算机网络编程**

**学院专业：计算机科学与工程学院**

**学生姓名：潘晨**

**学** **号：2018081308012**

**指导教师：向渝**

**日** **期：2020 年 12 月 21日**



**电 子 科 技 大 学**

**实 验 报 告**

**实验五**

|  |  |
| --- | --- |
| **学生姓名： 潘晨** | **学 号： 2018081308012** |
| **一、实验室名称： 主楼A2-412** | |
| **二、实验项目名称： 基于TCP协议的简单聊天软件的实现** | |
| **三、实验原理：**  1. TCP协议  TCP协议被用于应用程序之间的通信，当一个应用程序希望和另一个应用程序通信时，首先会发送一个通信请求（拥有固定的IP地址）。双方需进行三次握手（如下图所示）后，TCP将在两个应用程序之间建立全双工的通信。这个通信将占有双方之间的通信线路，直到一方关闭。    2. 程序模块构架  （1）客户软件  本次客户软件的编写涉及了多个函数模块，包括init、start、\*recv\_thread、main函数。这些函数模块有着各自的功能和相互调用关系，具体功能说明如下图所示：  （2）服务器端  本次服务器端软件的编写涉及了多个函数模块，包括init、SendAll、\*sever\_thread、sever和main函数。这些函数模块有着各自的功能和相互调用关系，具体功能说明如下图所示： | |
| **四、实验目的：**  通过本次对编写基于TCP协议的客户和服务器端软件程序，进一步加深课堂对TCP通信过程的理解，同时也通过具体实操锻炼代码能力和自主设计能力。主要部分为以下两点：   * 掌握TCP客户/服务器端软件的工作原理及通信过程 * 掌握针对字符流的TCP客户/服务端软件的编程步骤和编程实现，并最终实现两个客户端之间的字符流消息传送。 | |
| **五、实验内容：**  本次实验主要在Linux操作系统上利用C语言编写基于TCP协议的TCP客户与服务器端软件，实验内容包括以下几个步骤：   * 编程环境的准备，要求在Linux系统下安装GCC或CC编译器。 * 利用Vi编辑器，基于C语言编写TCP客户/服务器端软件的代码。 * 利用GCC编译器调试并形成可执行文件。 * 运行服务器和客户程序，验证功能是否实现。 | |
| **六、实验器材（设备、元器件）：**  系统：Ubuntu16.04.1  系统内核：Linux gpuserver06-SYS-7048GR-TR 4.15.0-126-generic  内存：65841760 kB  显卡：四块GeForce GTX 1080 Ti  连接工具：Windows远程桌面（mstsc）+zerotier（内网穿透） | |
| **七、实验步骤及操作：**  1. 连接服务器  Ctrl+R打开Windows运行框，输入mstsc启动远程连接服务并输入服务器IP。在弹出的界面输入账号和密码，若显示界面则表示连接成功。如下图所示：  2. 编写源程序  （1）服务器端软件  打开terminal终端，输入vi回车启动vi编辑器，输入i，进行编写程序（内容如下图所示），Esc键退出编辑模式，并输入:wq Sever.c保存为名为Sever的C语言文件并退出。    （2）客户端端软件  打开terminal终端，输入vi回车启动vi编辑器，输入i，进行编写程序（内容如下图所示），Esc键退出编辑模式，并输入:wq Client.c保存为名为Client的C语言文件并退出。    3. 编译和运行程序  （1）服务器端  利用GCC命令gcc -o Sever Sever.c – pthread（使用Linux线程模块需加上-pthread参数）将echo\_c.c编译生成名为Sever的可执行文件，。下图为gcc编译成功的截图及生成的echo\_c可执行文件。    （2）客户软件  利用GCC命令gcc -o Client Client.c - pthread（使用Linux线程模块需加上-pthread参数）将echo\_c.c编译生成名为Client的可执行文件，。下图为gcc编译成功的截图及生成的Client可执行文件。 | |
| **八、实验数据及结果分析：**  1. 结果展示  首先利用sudo ./Sever命令启动服务器端，开启聊天室，完成一些初始化功能。运行结果如下所示：  利用sudo ./Client命令开启两个客户程序，并输入相关的用户名，进入聊天室进行信息聊天。运行结果如下图所示：  服务器端信息    客户a界面    客户b界面  利用Ctrl+C终止服务器后，在用户a和用户b界面中输入信息，但双方都未收到相关信息。运行结果如下图所示：  服务器终止  客户a消息无法发出聊天  客户b无法聊天  2. 结果分析  如结果展示所示，在成功编译客户端和用户程序后，分别执行一个服务器端可执行文件和两个客户端可执行文件。两个客户端分别名为a和b，并且各自独立发送信息后，在聊天室和各自的用户界面上均有提示，表明聊天室程序运行良好。同时，当使用Ctrl+C关闭服务器端后，各用户的信息无法发出，也无法接收信息，进一步说明程序的正确性，实验效果与预期一致。 | |
| **九、实验结论：**  本次实验的任务是编写完成TCP的聊天室配套软件，包括可支持多人聊天的服务器端和聊天的用户端，在结果分析中可以发现不同的用户可以通过服务器这个平台进行实时聊天，并且随着聊天室（服务器端）的关闭，用户之间无法发送和接收信息。由此看出，本次实验较为出色地完成了目标任务。 | |
| **十、总结及心得体会：**  本次实验可以看做是在实验一和二（针对ECHO服务的服务器和客户端）的延伸，本次实验更贴近于实际，将服务器视为平台角色，用户端之间实现相互交流。本次实验对我的课程知识和专业技能上都有了较大帮助。  对于课程知识而言，通过前四次实验和本次实验，我对TCP的三次握手的机理和与UDP的不同之处有了更为深刻的体会。尤其在本次实验编写聊天室程序中，多设置了管理TCP连接的函数，从而实现平台的管理功能。  对于专业技术而言，我借此机会复习了大一学过的C语言知识，同时对Ubuntu系统的各种指令和操作掌握地更加熟练，代码编程能力也得到了一定的提升，受益匪浅。 | |
| **十一、对本实验过程及方法、手段的改进建议：**  本次实验是针对TCP协议的聊天室软件，建议可以多增加下功能，例如撤销消息等操作，增加更多的趣味性。 | |
| **报告评分：**  **指导教师签字：** | |