****

**网络程序设计课程设计**

**第1组**

**网工1801班**

**2020-2021-1学期**

|  |  |  |
| --- | --- | --- |
| **课程设计基本信息** | | |
| **题目** | **群聊天软件的设计与实现** | |
| **时间** | **2020年12月28日至2020年12月31日** | |
| **地点** | **2-16-504** | |
| **指导教师** | **梁荣** | |
| **组织方式** | **多人组队，合作完成** | |
| **小组成员信息及成绩** | | |
| **姓名** | **学号** | **成绩** |
| 吴斌 | **18408020129** |  |
| 刘涛 | **18408020130** |  |
| 高宇翔 | **18408020128** |  |
| - | - |  |
| - | - |  |
| - | - |  |
| **指导教师评语** | | |
|  | | |

**目 录**

[**1. 选题分析及设计目标 1**](#_Toc60260808)

[**2. 概要设计 1**](#_Toc60260809)

[**2.1 模块划分及功能描述 1**](#_Toc60260810)

[**2.2 模块间交互关系 3**](#_Toc60260811)

[**3. 详细设计 3**](#_Toc60260812)

[**3.1 函数设计 3**](#_Toc60260813)

[**3.2 数据结构设计 3**](#_Toc60260814)

[**3.3 函数调用关系 4**](#_Toc60260815)

[**4. 实现代码 4**](#_Toc60260816)

[**5. 测试报告 11**](#_Toc60260817)

[**6. 遇到的问题及解决方法 13**](#_Toc60260818)

[**7. 参考文献 14**](#_Toc60260819)

[**8. 个人小结 14**](#_Toc60260820)

[**8.1总结1（吴斌，18408020129，网络工程1801） 15**](#_Toc60260821)

[**8.2总结2（刘涛，18408020130，网络工程1801） 15**](#_Toc60260822)

[**8.3总结3（高宇翔，18408020128，网络工程1801） 17**](#_Toc60260823)

1. **选题分析及设计目标**

**选题分析：**选题为设计一个群聊软件。支持群聊天是当前通讯软件的必备功能。群聊天功能的实现依赖群成员的管理以及群消息的转发。要设计一个支持群聊天的应用协议（选做），并基于Linux平台进行编程实现。

本系统实现一个基于Linux的模拟即时通信系统，

要求实现以下功能：  
1、模拟即时通信系统可以实现多人同时在线聊天功能；  
2、在线聊天用户登录本系统需输入用户名和密码；  
3、本系统需能够查询历史聊天记录；  
4、本系统运行后，需启动一个守护进程，该守护进程记录本系统启动和关闭的时间，每个用户登录和退出时间；

该聊天工具是在Linux平台下进行开发与实现的，可以实现多人通信和交流。系统采用了TCP/IP协议和socket接口。从总体来讲，该聊天系统主要包括服务端与客户端。

**具体实现：**群聊天应用的实现依赖两部分基础的功能：一是群成员的动态维护；二是群消息的转发。可以基于客户端/服务器模型进行实现。客户端向服务器注册加入群组，服务器进行群成员的动态维护；客户端发送消息给服务器，服务器负责向群内成员进行转发。

**设计目标：**

(1) 设计一套支持群聊天应用的协议，能够支持群的建立、群成员的加入、退出以及群消息的转发。

(2) 设计支持群聊天应用的关键数据结构、算法和代码，实现群聊天应用。

1. **概要设计**

## 2.1 模块划分及功能描述

**模块划分：**

系统应实现4大功能:

1.注册与登录系统2.私聊3.公聊4.文件传输

（文件传输、保存聊天记录、群组的建立、加入特定群组选作，不强制要求。）

服务端功能实现模块：  
1.用户通过用户名和密码进行登录(选作)  
2.接收和发送用户聊天信息功能（接受发信用户发来的信息，再将信息转发。  
客户端功能实现模块：  
1.验证登录用户的信息  
2.客户端之间进行信息传递交流  
3.保存聊天记录

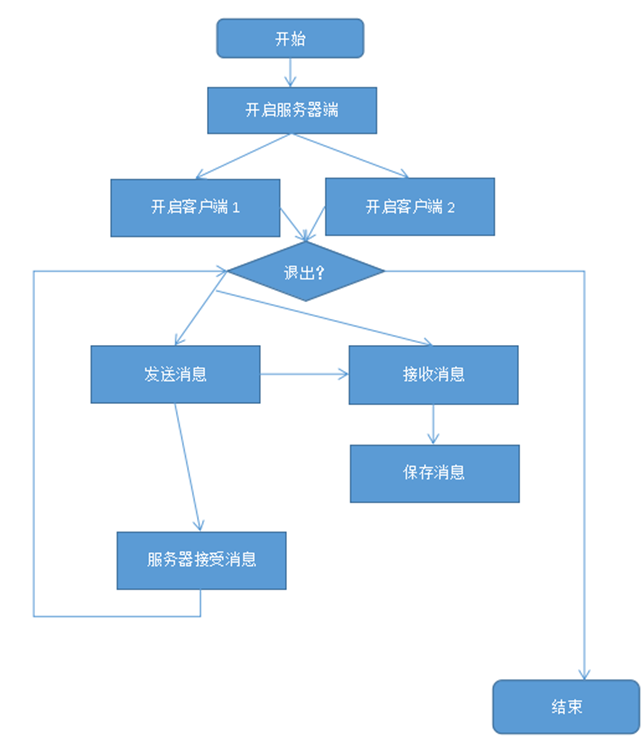
**功能描述：**

1.**注册与登录系统。**客户端即用户需要注册后登录才可进行群聊客户端把自己的标识符即用户名和密码发给服务器后，由服务器进行检测和比对，只有当其与服务器中已有的数据库（或结构体）里比对成功时，才可成功登录进行群聊。注册即客户端在没有用户名和密码的情况下，通过想服务器发送请求来获取一个合法的用户名和密码的功能。

2.**私聊。**即用户可先向服务器查询在线用户，然后可通过意愿向某一位用户进行一对一的数据通信。在使用udp套接字时可不必知道目的客户端的全部信息。

**3.公聊。**用户向服务器发送信息，然后由服务器进行对特定分组或者广播的形式进行转发。即一人发送多人接受的通信模式。

**4.文件传输。**用户可在私聊的基础上进阶，对某一位在线用户进行文件的传输。

****

**（图为流程图）**

**2.2 模块间交互关系**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 注册登录模块 | 公聊模块 | 私聊模块 | 文件传输模块 |
| 用户注册 | **√** |  |  |  |
| 用户登录 | **√** |  |  |  |
| 系统广播 |  | **√** |  |  |
| 上下线提示 |  | **√** |  |  |
| 用户私聊 |  |  | **√** |  |
| 文件传输 |  |  |  | **√** |
| 服务器踢出用户 |  | **√** |  |  |

1. **详细设计**

**3.1 函数设计**

Client端（客户端）：

Main函数：程序的入口，调用其他函数的地方，是程序的起点和终点。

Snd函数： 负责向服务器发送信息以便服务器处理

Rcv函数： 负责从服务器读取信息（其他客户端的群发信息）

Server端（服务器）：

Main函数：程序的入口，调用其他函数的地方，是程序的起点和终点。

Quit函数： 被创建用于管理服务器的程序

Rev\_snd函数：被创建出来用于信息的接收和发送

**3.2 数据结构设计**

需要用到数据结构的地方我一一列举出来：

首先分析服务器需要一个数组结构体来存储每个客户端的文件描述符（即调用accept函数的返回值）、每个客户端的id（可理解为连接上服务器的先后顺序）、用户名、密码等等。以便在私聊模块以及文件传输模块的时候用户们可以利用服务器来转发私聊信息。

其次需要一个数组来存储客户端的文件描述符以便在群发模块时，方便分组然后进行群组转发。

接下来结合代码细化分析：

Server端：用到struct client[maxclinum];来存储连接上服务器端的属性例如文件描述符，name，密码（如果采用登录验证系统的话）；

struct client{int pthid;char name[10];}; //结构体定义完成

struct client client[10];int connfd[MAXMEM];

struct sockaddr\_in serv\_addr, cli\_addr;

Client端：

char buf[BUFFSIZE]; char name[32], buf[BUFFSIZE];

其次，struct sockaddr\_in serv\_addr。客户端的数据结构用到的较少。

**3.3 函数调用关系**

客户端：

main函数调用其内实现的save函数，保存消息记录。并且在该函数内调用用于socket通信的接口实现通信编程，如：inet\_addr，用于接收服务器端口地址信息，建立连接；htons接收指定的服务端口。Main函数：程序的入口，调用其他函数的地方，是程序的起点和终点。Snd函数： 负责向服务器发送信息以便服务器处理Rcv函数： 负责从服务器读取信息（其他客户端的群发信息）

服务器端：

调用socket接口实现通信，其中bind函数用于信息地址的绑定，建立信息的连接；listen函数用于监听客户端的状态，便于信息的连接。Main函数：程序的入口，调用其他函数的地方，是程序的起点和终点。Quit函数：被创建用于管理服务器的程序Rev\_snd函数：被创建出来用于信息的接收和发送。

1. **实现代码**

该课题需要至少俩个源码，一个server端，一个client端。（不包含自写头文件的情况下.）

Server端代码：ser.c

#include <stdio.h>

#include <stdlib.h> // exit函数的头文件

#include <string.h>

#include <unistd.h> // bind listen的头文件

#include <time.h> // time(NULL)，ctime(&ticks)获取系统时间的头文件

#include <netinet/in.h>

#include <arpa/inet.h> // 必须包含，用于inet\_ntop

#define PORT 8000 //使用特定的端口

#define MAXMEM 10 //规定特定的最大连接用户数目

#define BUFFSIZE 128 //设置最大

int listenfd, connfd[MAXMEM];

struct client{int pthid;char name[10];}; //结构体定义完成

struct client client[10]; //定义一个数组结构体变量，大小为10

void quit(); //声明quit函数

void rcv\_snd(int p); //声明snd函数

int main(){ struct sockaddr\_in serv\_addr, cli\_addr; //定义一个服务器端，一个客户端

int i;time\_t ticks; //定义一个常量i方便以后使用for循环

pthread\_t thread; //定义一个线程id常量

char buff[BUFFSIZE]; //定义一个字符数组用来存储数据

printf("running...\n(Prompt: enter command ""quit"" to exit server)\n"); //服务器端的提示信息

bzero(&serv\_addr, sizeof(struct sockaddr\_in));

serv\_addr.sin\_family = AF\_INET;

serv\_addr.sin\_port = htons(PORT);

serv\_addr.sin\_addr.s\_addr = htonl(INADDR\_ANY); // 初始化serv\_addr

listenfd = socket(AF\_INET, SOCK\_STREAM, 0);

if(listenfd < 0){perror("fail to socket");exit(-1);}

int opt=SO\_REUSEADDR;

setsockopt(listenfd,SOL\_SOCKET,SO\_REUSEADDR,&opt,sizeof(opt)); //监听套接字

if(bind(listenfd, (struct sockaddr \*)&serv\_addr, sizeof(serv\_addr)) < 0){perror("fail to bind");exit(-2);}

listen(listenfd, MAXMEM);// 规定最大的监听数

pthread\_create(&thread, NULL, (void \*)(quit), NULL); //创建一个线程，对服务器程序进行管理

for(i=0; i<MAXMEM; i++)connfd[i] = -1; // 将套接字描述符数组初始化为-1，表示空闲

while(1){

int len;

for(i=0; i<MAXMEM; i++){if(connfd[i] == -1)break;} //找到未使用的最小的i值

connfd[i] = accept(listenfd, (struct sockaddr \*)&cli\_addr, &len); // accept 从listen接受的连接队列中取得一个连接

client[i].pthid = connfd[i]; //在特定的线程内给结构体中的pthid赋值

if(connfd[i] < 0){perror("fail to accept");}// 此句可以不用，accept会阻塞等待

ticks = time(NULL);

printf("%.24s\n\tconnect from: %s, port %d\n",

ctime(&ticks), inet\_ntop(AF\_INET, &(cli\_addr.sin\_addr), buff, BUFFSIZE),

ntohs(cli\_addr.sin\_port)); // inet\_ntop的使用以及调用时间函数的使用

pthread\_create(malloc(sizeof(pthread\_t)), NULL, (void \*)(&rcv\_snd), (void \*)i); //针对当前套接字创建一个线程，对当前套接字的消息进行处理

} //while函数结束

return 0;

} //main函数结束

void quit(){ //设置quit函数

int choose,pid,i;

char msg[10];

while(1){

printf("Hi!欢迎进入群组聊天室管理台\n\n");

printf("请输入您想实现的功能\n"); //实现服务器的菜单选项，服务器选择功能模块

printf("1.显示在线客户端\n");

printf("2.踢出指定客户端\n");

printf("3.关闭服务器\n");

printf("聊天管理台启动成功，等待用户进入聊天室......\n");

scanf("%d",&choose);

switch(choose){

case 1: for(i=0;i<MAXMEM;i++)

if(connfd[i] != -1)

printf("%d\t",connfd[i]);

printf("\n");

break;

case 2: printf("请输入指定客户端的套接字");

scanf("%d",&pid); //获取想踢出客户端的pid

for(i=0;i<MAXMEM;i++){

if(connfd[i]==pid)

{shutdown(pid,SHUT\_RDWR);connfd[i] = -1;} //if语句结束

} //for循环结束

break; //case结束

case 3: printf("请输入quit\n");

scanf("%s", msg); // scanf 不同于fgets, 它不会读入最后输入的换行符

if(strcmp(msg, "quit") == 0){printf("Byebye... \n");close(listenfd);exit(0);} //if语句结束

break; //case结束

} //switch语句结束

} //while循环结束

} //quit函数定义结束

void rcv\_snd(int n){

int len, i,j;int ret,pid;

char name[32], mytime[32], buf[BUFFSIZE];

time\_t ticks;

write(connfd[n], "your name: ", strlen("your name: ")); // 获取此线程对应的套接字用户的名字

len = read(connfd[n], name, 32);

if(len > 0)

name[len-1] = '\0'; // 去除换行符

for(i=0;i<10;i++){

if(!strcmp(client[i].name,name)){

write(connfd[n], "sorry 因用户名已被注册已自动退出 请尝试重新注册", strlen("sorry 因用户名已被注册已自动退出 请尝试重新注册"));

//进行用户名之间的对比，若重复则会退出后进入的客户端

shutdown(connfd[n],SHUT\_RDWR);

connfd[n] = -1;} //if语句结束

} //for循环结束

strcpy(client[n].name,name);

strcpy(buf, name);

strcat(buf, "\t进入了群聊...\n\0");

pid=connfd[n];// 把当前用户的加入 告知所有用户

for(i=0; i<MAXMEM; i++){

if(connfd[i] != -1)

write(connfd[i], buf, strlen(buf));

} //for循环结束

while(1){

char temp[BUFFSIZE];

if((len=read(connfd[n], temp, BUFFSIZE)) > 0){

temp[len-1] = '\0'; // 当用户输入bye时，当前用户退出

if(strcmp(temp,"show")==0){

for(i=0;i<MAXMEM;i++)

if(connfd[i] != -1){

char chuan;

chuan=connfd[i]+'0';

write(connfd[n], &chuan, sizeof(chuan));

} //for循环完成

} //if完成

if(strcmp(temp, "bye") == 0){

printf("%s 下线了...\n",name);

close(connfd[n]);

connfd[n] = -1;

pthread\_exit(&ret);}

ticks = time(NULL);

sprintf(mytime, "%.24s\r\n", ctime(&ticks));

strcpy(buf, name);

strcat(buf, "\t");

strcat(buf, mytime);

strcat(buf, "\r\t");

strcat(buf, temp);

strcat(buf, "\n");

if(!strncmp(temp,"all",3))

{

for(i=0; i<MAXMEM; i++)

{

if(connfd[i] != -1)

write(connfd[i], buf, strlen(buf));

}

}

if(strncmp(temp,"to",2)==0)

{

for(i=0;i<10;i++)

{ char p[5] = "to" ;

strcat(p,client[i].name);

if(strncmp(p,temp,5)==0)

{write(connfd[i], buf, strlen(buf));}

}}}

}}

Client端源码：cli.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <netinet/in.h>

#define BUFFSIZE 128

#define HOST\_IP "127.0.0.1"

#define PORT 8000

int sockfd;

void snd();

void rcv();

int main(){

pthread\_t thread; // pthread\_t 线程，gcc编译时需加上-lpthread

struct sockaddr\_in serv\_addr; // struct sockaddr\_in

char buf[BUFFSIZE];

bzero(&serv\_addr, sizeof(struct sockaddr\_in)); // bzero 清零

serv\_addr.sin\_family = AF\_INET; // sin\_family AF\_INET

serv\_addr.sin\_port = htons(PORT); // sin\_port htons(PORT)

inet\_pton(HOST\_IP, &serv\_addr.sin\_addr); //初始化服务端地址结构

sockfd = socket(AF\_INET, SOCK\_STREAM, 0); // 创建客户端套接字

if(sockfd < 0){perror("fail to socket");exit(-1);}

printf("connecting... \n");

if(connect(sockfd, (struct sockaddr \*)&serv\_addr, sizeof(serv\_addr)) < 0){perror("fail to connect");exit(-2);} // 与服务器建立连接

/\* ===程序分做两个线程 === \*/

pthread\_create(&thread, NULL, (void \*)(&snd), NULL); //创建发送消息的线程，调用发送消息的函数snd

printf("请选择您想实现的功能\n\n");

printf("This is a user info\n");

printf("显示在线客户端,请输入'show'\n");

printf("发送消息,选择使用的功能，输入all为群发，to+您想发送的用户名为私发\n");

printf("建议在all或者to+用户名后加上一个空格方便信息的查看\n");

printf("退出该客户端的连接请输入bye\n");

while(1){

int len;

if((len=read(sockfd, buf, BUFFSIZE)) > 0){ // read 读取通信套接字

buf[len] = '\0'; // 添加结束符，避免显示缓冲区中残留的内容

printf("\n%s", buf);

fflush(stdout);} // if语句结束，fflush 冲洗标准输出，确保内容及时显示

} //while语句结束

return 0;

} //main函数结束

void snd(){ // 发送消息的函数

char name[32], buf[BUFFSIZE];

fgets(name, 32, stdin); // fgets 会读取输入字符串后的换行符

write(sockfd, name, strlen(name)); // write 写入通信套接字

while(1){

fgets(buf, BUFFSIZE, stdin);

write(sockfd, buf, strlen(buf));

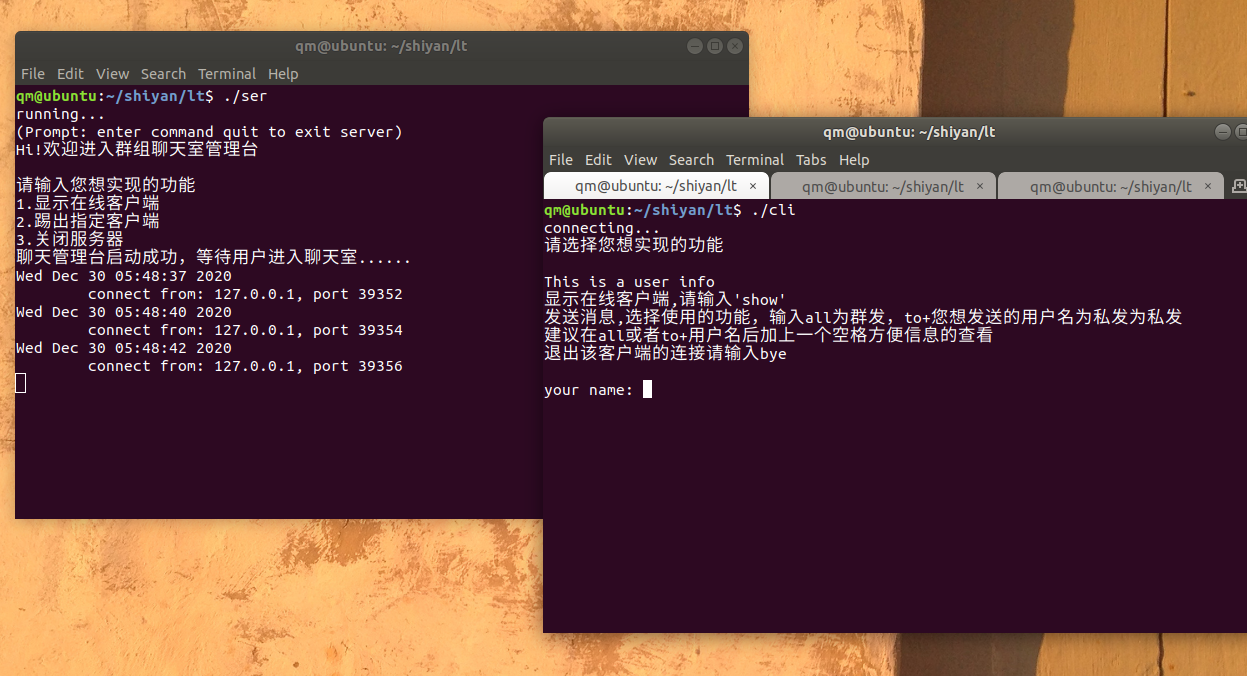
if(strcmp(buf, "bye\n") == 0) // 注意此处的\n

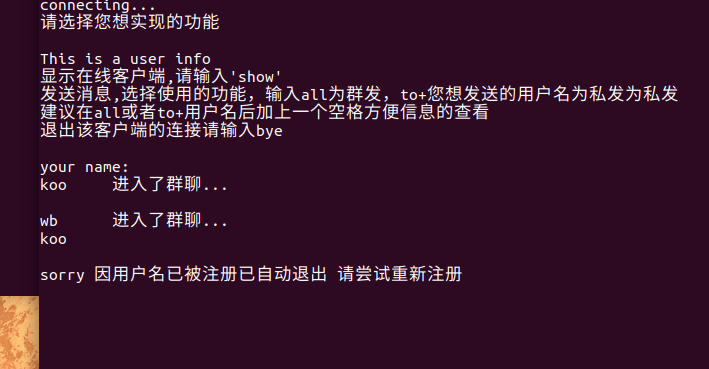
exit(0);} //while结束

} //函数定义完成

1. **测试报告**

放几张编写过程中以及最终成果的截图:（图解在图片下方）

**（首先是ser和cli运行出来的结果）**



**（图为用户名已被注册是会强制下线）**

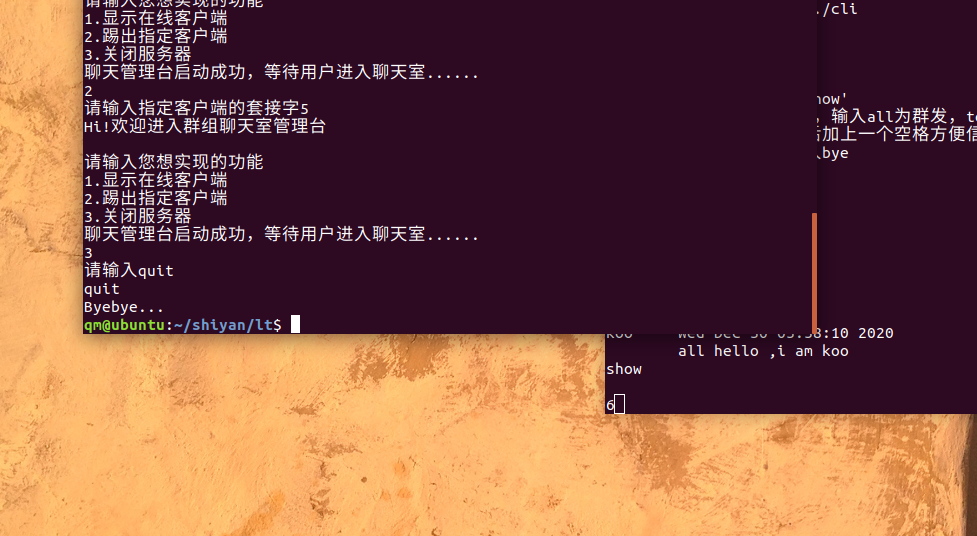
**（图为群发消息）**



**（图为私发信息，只有指定名字的用户才会收到）**



**（图为一个连续操作后，ser和cli的各自处理情况达到预期目标）**



**（图为关闭服务器）**

1. **遇到的问题及解决方法**

问题来源于我们组其实在听说了老师说简单以后讨论了一下觉得实现应该不会太难，就决定三个成员分别用多进程、多线程、io复用分别先把框架编写好，随后讨论各自出现的问题，并最终决定采取那种方案。

问题①： 在循环嵌套中，结构上出了问题，各种报错，各种编译出来却出bug，各种警告，各种变量错误。

解决办法：没什么好的解决办法，重新捋清思路，捋清结构，捋清功能。一步一步看，一步一步循环。一步一步找，一步一步做。也是功夫不负有心人吧，最后成功解决。

问题②： 在编写代码进行read（）和write（）俩个函数的数据通信时，出现了不少问题。

解决办法： 也没什么好的解决办法，一组一组（一组即为一个read和一个write）写，一组一组输入输出测试。

问题③： 其实一开始也没打算组太多人做这个课题，又加上赶上放元旦假期。弄到真正动手做的时候发现人手和各自能力以及时间都有些不够。任务功能模块较多较杂。

解决方法：没办法，熬夜“加班”写呗，本来就少一天时间，晚上赶赶工。

问题④： 在使用最终方案多线程时，出现了服务器收到的客户端信息无法与给定的协议前缀比对成功的问题。

解决方法：最终查阅资料后使用了比strcmp更适合的函数strncmp；只进行给定固定前缀的字符串对比。最终成功解决问题。

1. **参考文献**

①：网络资源（含各大社区一些博客，以及一些百科）。

②：本学期网络实验设计书《网络程序设计实验指导书》的部分讲解。

③：本学期教材《linux/unix网络编程》上的各个函数。

④：张雪坤.基于Socket的网络聊天程序设计与实现.电脑编程技巧与维护,

⑤：韩利凯.利用SOCKET进行网络通信程序设计.西安联合大学学报

1. **个人小结**

**8.1总结1（吴斌，18408020129，网络工程1801）**

本次历时三天的夜以继日的调试和编写工作，打心眼里说真的成长许多，毕竟学校以及老师开设课程设计这个项目就是为了让学生们有所成长。所以也算是不负众望吧哈哈。

首先各种调试各种编译错误逻辑错误必不可少，因为我们组因为一致讨论以后觉得不算太复杂以后就决定三种并发方式一人一种最后再决定最终方案，也正好我们组三位成员嘛。所以这次课设真的是算是独自面对一个难题。从头开始，从tcp套接字开始编写；从socket、bind、listen开始编写。算是"呕心沥血"。

开头说夜以继日其实没有夸大太多。早上起床我们哥几个就开电脑开写，并且时不时开个小会讨论一下各自遇到的恶心问题，然后吐槽一下运行函数过程中的不协调。

这其实是对团队工作的一种理解和趣味。为以后的工作以及生活的团队合作内容打下夯实基础。然后说说在本次课设活动中获得的东西以及一些失去的东西吧。

首先，肯定又获得了一些学业性的东西，比如一些函数的正确使用方法和学会了一些以前没使用过的函数例如strncmp函数，该函数是strcmp的一个变通，它是在比对字符串的时候只比对一个固定前缀的函数。若前若干位都比对成功，则返回值为0。该函数是最后收尾工作时找了很多资料发现的，真的是解决了我们组的很多问题。在解决了问题的基础上，有了更加坚决的决心去学习更多的函数运用，有了更浓厚的兴趣去深入学习网络知识。

我觉得这是值得的。

**8.2总结2（刘涛，18408020130，网络工程1801）**

通过本次的课程设计，我们小组成功地实现了多个客户在一起进行群聊，以及两个客户端之间进行私发消息的功能，并在此基础上完善了一些功能，使用户的体验得以提升。我在这次的课程设计中学到了很多，发现了像交流能力的不足，表达能力的欠缺，收集资料能力不足等等问题。在这次的课设中我都尽力去克服这些困难，努力配合集体一起完成这次的课程设计。

我收获最大的方面还是在知识的巩固上面，这次的课设用到了很多之前学习过的内容，像数据结构和c语言等基础知识，我接着这次的课设也复习到了一些知识。对于这次用到的多线程我在学习理论知识时并没有在这上面画上很多的功夫，但通过这次的课设，我体会到了多线程在程序设计中的便捷，它不像多进程那样具有复杂的进程通信，这为程序的产生提供了很大的便捷。希望以后能有机会多做一些这样的程序。

## 8.3总结3（高宇翔，18408020128，网络工程1801）

这次课程设计经过三天之后总算是做完了，程序，差不多要完成的功能基本都实现了。在做的过程中出现过几次小问题，也花费了一些时间才做通的。  
 在本次课程设计的过程中，学习了很多有关的知识。这样的项目对我学过的网络编程课程是一个综合性很高的实践。一些以前没有学得很杂实的课程的内容，由于需要在实践中运用，刚开始我也感到很头痛。但回过头再去看教科书，经过一段时间的钻研，对与这些知识点的相关的背景，概念和解决方案理解得更透彻了，已经差不多熟悉了课程设计的过程。  
 通过这次实验，我也了解到了网络编程的博大精深，我做的也只是一些皮毛之类的东西，要想真正学好网络编程，搞开发的话，那确实还差得太远了，这次这个实验在通过三天的精心准备后，终于完成了，虽然过程中是很痛苦的，因为很多东西搞得不是太懂，向同学请教了，感谢他们的帮助让我完成这个实验的负责模块，同时我也上网查了不少东西，经过慢慢琢磨，终于把这个程序搞出来了，不过其中也暴露出了我的一些弱点，就是实践能力还比较欠缺，对教材和老师的依赖过多，以后要更多的靠自己琢磨，我相信只要有心学好这门课，在难的东西也会变得简单的。  
 另外我还充分体会了从事编程方面的工作需要特别严谨认真的态度和作风，一点都马虎不得。每一个细微的细节都必须十分的注意，如果不认真思考决策，就会出现或大或小的错误，如果早期的错误隐藏下来，对后面的工作影响就会很大，甚至有时要推倒很多前面做的工作重来。有时候，我自己觉得我写的程序非常的正确，但是就是编译通不过，在查找错误的过程中，需要很认真的去看自己到底哪里写错了，有时候一个错误就要找半天，十分费劲。  
    总的来说这次课设完成的相对较为成功，这离不开组内其他成员的辛苦工作。