```
#include "client.h"
char myname[20];     //用于保存本地名字
char signname[40];   //用于保存个性签名
char mylocalname[20]; //用于保存传输文件来源的名字
char fsignname[40]; //用于保存文件传输名字
/*********************************网络连接**************************/

void expression(char name[],char msg[])
{
	if(strcmp(msg,"/xl") == 0)
	{
		sprintf(msg,"表情：%s 做了个笑脸 ☺:-D",name);
	}
	if(strcmp(msg,"/js") == 0)
	{
		sprintf(msg,"表情：%s 很沮丧 ☹ :-(",name);
	}
	if(strcmp(msg,"/jy") == 0)
	{
		sprintf(msg,"表情：%s 很惊讶 :-O",name);
	}
	if(strcmp(msg,"/hh") == 0)
	{
		sprintf(msg,"表情：哈哈 ^_^");
	}
	if(strcmp(msg,"/kkl") == 0)
	{
		sprintf(msg,"表情：%s 快哭了 T_T",name);
	}
	if(strcmp(msg,"/zk") == 0)
	{
		sprintf(msg,"表情：%s 抓狂 >_<",name);
	}
	if(strcmp(msg,"/mmj") == 0)
	{
		sprintf(msg,"表情：喵喵叫 (=^_^=)");
	}
	if(strcmp(msg,"/yd") == 0)
	{
		sprintf(msg,"表情：晕 (×_×)");
	}
	if(strcmp(msg,"/zt") == 0)
	{
		sprintf(msg,"表情：猪头 ^(oo)^");
	}
```

```
    if(strcmp(msg,"/dgx") == 0)
    {
        sprintf(msg,"表情：大狗熊  (￣(工)￣)");
    }
    if(strcmp(msg,"/bz") == 0)
    {
        sprintf(msg,"表情：闭嘴吧你 :-x");
    }
}


void handle_password(char *pass) //用*代替密码
{
    int i=0;
    system("stty -icanon"); //设置一次性读完操作，即 getchar()不用回车也能获取字符
    system("stty -echo");    //关闭回显，即输入任何字符都不显示
    while(i < 16)
    {
        pass[i]=getchar();                          //获取键盘的值到数组中
        if(i == 0 && pass[i] == BACKSPACE)
        {
            i=0;                //若开始没有值，输入删除，则，不算值
            pass[i]='\0';
            continue;
        }
        else if(pass[i] == BACKSPACE)
        {
            printf("\b \b");//若删除，则光标前移，输空格覆盖，再光标前移
            pass[i]='\0';
            i=i-1;                          //返回到前一个值继续输入
            continue;                       //结束当前循环
        }
        else if(pass[i] == '\n')            //若按回车则，输入结束
        {
            pass[i]='\0';
            break;
        }
        else
        {
            printf("*");
        }
        i++;
    }
    system("stty echo");      //开启回显
    system("stty icanon");    //关闭一次性读完操作，即 getchar()必须回车也能获取字符
```

```
}


/* 连接服务器 */
int Connect_Server(void)
{
	int client_socket = socket(AF_INET, SOCK_STREAM, 0);
	if(client_socket == -1)
	{
		perror("socket error");
		return -1;
	}
	struct sockaddr_in addr;
	memset(&addr, 0, sizeof(addr));
	addr.sin_family = AF_INET;
	addr.sin_port = htons(PORT);
	inet_aton("127.0.0.1",&(addr.sin_addr));

	int ret = connect(client_socket,(struct sockaddr *)&addr, sizeof(addr));
	if(ret == -1)
	{
		perror("connect error");
		return -1;
	}
	printf("成功连接到服务器 : %s\n",inet_ntoa(addr.sin_addr));

	return client_socket;
}

/* 访问服务器 主界面 */
int Ask_server(int client_socket)
{
	char ch;
	int ret;
	while(1)
	{
		main_menu();
		printf("请输入您要做操作\n");
		scanf("%c",&ch);
		while(getchar() != '\n');
		switch(ch)
		{
			case '1':	//注册
				regis(client_socket);
```

```
                    break;
                case '2':       //登录
                    ret = entry(client_socket);
                    if (ret == 1)
                    {
                      User_used(client_socket); //调用函数表示用户界面
                    }
                    break;
                case '3':       //退出
                    exit(0);
                    break;
            }
        }
}

/*********************************界面*****************************/

/* 客户端主界面 */
void main_menu(void)
{
    system("clear");
    printf("\n\n\n\n\n\n\n\n");
    printf("+++++++++++++++++++++++++++++++++++++++++++++\n");
    printf("+++\t      欢迎使用安工大聊天室          +++\n");
    printf("+++\t\t1,注册                        +++\n");
    printf("+++\t\t2,登录                        +++\n");
    printf("+++\t\t3,退出                        +++\n");
    printf("+++++++++++++++++++++++++++++++++++++++++++++\n");

}

/* 用户登录界面 */
void user_menu(void)
{
    system("clear");
    printf("%s: %s\n",myname,signname);
    printf("\n\n\n\n\n\n\n");
    printf("+++++++++++++++++++++++++++++++++++++++++++++\n");
    printf("+++                                     +++\n");
    printf("+++\t    欢迎使用安工大聊天室          +++\n");
    printf("+++                                     +++\n");
    printf("+++++++++++++++++++++++++++++++++++++++++++++\n");
    printf("+++\t\t1,群聊                        +++\n");
    printf("+++\t\t2,私聊                        +++\n");
    printf("+++\t\t3,退出登录                    +++\n");
```

```
        printf("+++\t\t4,查看聊天记录          +++\n");
        printf("+++\t\t5,显示当前在线人员      +++\n");
        printf("+++\t\t6,修改个性签名          +++\n");
        printf("+++\t\t7,修改密码              +++\n");
        printf("+++\t\t8,传输文件              +++\n");
        printf("++++++++++++++++++++++++++++++++++++++++++++\n");
}

/*******************************主界面功能************************/
/* 注册账号 */
void regis(int client_socket)
{
        Msg msg;
        msg.cmd = 1;
        printf("注册,请输入账号名: ");
        scanf("%s",msg.fromname);
        while(getchar() != '\n');
        printf("注册,请输入密码: ");
        scanf("%s",msg.password);
        while(getchar() != '\n');
        write(client_socket, &msg, sizeof(msg));

        read(client_socket, &msg, sizeof(msg));
        if(msg.cmd == 1001)
        {
                printf("注册成功\n");
        }
        else if (msg.cmd == -1)
        {
                printf("用户名以存在,注册失败\n");
        }
        else
        {
                printf("系统繁忙,注册失败\n");
        }

        sleep(2);
}

/* 登录账号 */
int entry(int client_socket)
{
        Msg msg;
        msg.cmd = 2;
        int i_password=0;
```

```
    char *ch_password;//用于暂存密码

    printf("登录,请输入账号名: ");
    scanf("%s",msg.fromname);
    while(getchar() != '\n');

    printf("登录,请输入密码: ");

    handle_password(msg.password);

    printf("\n");

    write(client_socket, &msg, sizeof(msg));

    read(client_socket, &msg, sizeof(msg));
    if(msg.cmd == -1)       //表示用户不存在
    {
        printf("登录失败,用户不存在.\n");
        sleep(2);
        return -1;
    }
    if(msg.cmd == -2)
    {
        printf("登录失败,密码错误.\n");
        sleep(2);
        return -2;
    }
    if(msg.cmd == 1002)
    {
        printf("登录成功,登录中...\n");
        strcpy(myname,msg.fromname);          //保存在线名字
        strcpy(signname, msg.signname);     //保存个性签名
        sleep(2);
        return 1;
    }
}

/*********************************用户界面功能**********************/

/* 用户界面 */
void User_used(int client_socket)
{
    /* 要进行读写分离 */
    pthread_t read_id;
    pthread_create(&read_id, NULL, readMsg, (void *)client_socket);
```

```
pthread_detach(read_id);     //等待线程分离

char ch;
int i = 1;
while(i)
{
        user_menu();
        printf("请输入您要做操作\n");
        scanf("%c",&ch);
        while(getchar() != '\n');
        switch(ch)
        {
                case '1':               //群聊
                        chat_all(client_socket);
                        break;
                case '2':               //私聊
                        chat_one(client_socket);
                        break;
                case '3':               //退出登录
                        entry_out(client_socket);
                        i = 0;
                        break;
                case '4':               //查看聊天记录
                        look_chat();
                        break;
                case '5':               //显示当前在线人数
                        see_now_time(client_socket);
                        break;
                case '6':               //修改个性签名
                        revise_sign(client_socket);
                        break;
                case '7':               //修改密码
                        revise_password(client_socket);
                        break;
                case '8':               //传输文件
                        transfer_file(client_socket);
                        break;
                case 'y':               //表示愿意接受文件
                        transfer_file_y(client_socket);
                        break;
                case 'n':               //表示不愿意接受文件
                        transfer_file_n(client_socket);
                        break;
        }
}
```

```
}

/* 读写分离 */
void * readMsg (void *v)
{
	int client_socket = (int)v;
	Msg buf;
	int i = 0;
	while(1)
	{
		bzero(&buf,sizeof(buf));
		int ret = read(client_socket, &buf, sizeof(Msg));
		if(ret == -1)
		{
			perror("read");
			break;
		}
		switch(buf.cmd)
		{
			case 3:	//群聊
				expression(buf.fromname,buf.msg);
				printf("收到了一条消息: %s\n",buf.msg);
				save_Chat(&buf);	//保存聊天记录
				break;
			case 4:	//私聊
				expression(buf.fromname,buf.msg);
				printf("%s 给你发了一条消息: %s\n",buf.fromname,buf.msg);
				save_Chat(&buf);	//保存聊天记录
				break;
			case -3:	//私聊失败
				printf("私聊失败,用户不存在或下线\n");
				break;
			case 5 :	//退出登录
				printf("%s 退出登录\n",buf.fromname);
				sleep(1);
				pthread_exit(NULL);	//线程退出
				break;
			case 6 :	//显示当前在线人数
				printf("当前在线人员:\n");
				printf("%s\n",buf.msg);
				break;
			case 7 :	//修改个性签名成功
				strcpy(signname,buf.signname);
				printf("修改个性签名成功\n");
				break;
```

```
                case -7 :   //修改个性签名失败
                    printf("修改个性签名失败\n");
                    break;
                case 8    :    //修改密码成功
                    printf("修改密码成功\n");
                    break;
                case -8 :       //修改密码失败
                    printf("修改密码失败\n");
                    break;
                case 9   :    //调用函数确认是否接受文件
                    system("clear");
                    printf("  请  问  你  是  否  接  受  来  自      %s    的  文
件 %s(y/n)\n",buf.fromname,buf.signname);
                    strcpy(mylocalname,buf.fromname);    //保存传输文件来源名字
                    sleep(1);
                    break;
                case -9 :    //表示传输文件失败,没有找到该人
                    printf("发送文件失败,好友不在线或不存在\n");
                    break;
                case 10 :    //表示愿意接受文件,开始传输
                    start_transfer_file(client_socket);
                    break;
                case -10 :   //表示不愿意接受文件
                    printf("发送文件失败,后又拒绝接受文件\n");
                    break;
                case 11 :    //接受文件
                    save_transfer_file(&buf);
                    i++;
                    break;
                    /* printf("%d\n",i); */
                default:break;
            }
        }
}

/* 群聊 */
void chat_all(int client_socket)
{
    Msg msg;
    msg.cmd = 3;
    strcpy(msg.fromname,myname);
    strcpy(msg.localname,"All");
    printf("请输入你要群发送的信息\n");
    scanf("%s",msg.msg);
    while(getchar() != '\n');
```

```
        write(client_socket, &msg, sizeof(Msg));
        sleep (2);
}

/* 私聊 */
void chat_one(int client_socket)
 {
        Msg msg;
        msg.cmd = 4;
        printf("请输入你要聊天的对象:\n");
        scanf ("%s",msg.localname);
        while(getchar() != '\n');

        printf("请输入要发送的内容: \n");
        scanf("%s",msg.msg);
        while(getchar() != '\n');

        strcpy (msg.fromname,myname);
        write(client_socket, &msg, sizeof(Msg));

        save_Chat(&msg);        //保存聊天记录

        sleep(2);
}

/* 退出登录 */
void entry_out(int client_socket)
{
        Msg msg;
        msg.cmd = 5;
        strcpy (msg.fromname,myname);
        write(client_socket, &msg, sizeof(Msg));
}

/* 查看聊天记录 */
void look_chat(void)
{
        Msg msg;
        strcpy(msg.fromname,myname);
        see_chat(&msg);
}

/* 显示当前在线人数 */
void see_now_time(int client_socket)
{
```

```
	Msg msg;
	msg.cmd = 6;
	strcpy (msg.fromname,myname);
	write(client_socket, &msg, sizeof(Msg));

	sleep(2);
}

/* 修改个性签名 */
void revise_sign(int client_socket)
{
	Msg msg;
	msg.cmd = 7;
	printf("请输入新的个性签名:");
	scanf("%s",msg.signname);
	while(getchar() != '\n');

	strcpy(msg.fromname,myname);	//需要保存名字
	write(client_socket, &msg, sizeof(Msg));

	sleep(2);
}

/* 修改密码 */
void revise_password(int client_socket)
{
	Msg msg;
	msg.cmd = 8;
	printf("请输入新的密码: ");
	scanf("%s",msg.password);
	while(getchar() != '\n');

	strcpy(msg.fromname,myname);	//需要保存名字
	write(client_socket, &msg, sizeof(Msg));

	sleep(2);
}


/* 传输文件 */
void transfer_file(int client_socket)
{
	Msg msg;
	msg.cmd = 9;
	printf("请输入你要传输文件的对象:");
```

```
        scanf ("%s",msg.localname);
        while(getchar() != '\n');

        printf("请输入你要传输的本地文件名:");
        scanf ("%s",msg.signname);
        while(getchar() != '\n');

        strcpy(msg.fromname,myname);     //需要保存名字
        write(client_socket, &msg, sizeof(Msg));

        printf("等待验收中...\n");

        strcpy(fsignname,msg.signname);
        strcpy(mylocalname,msg.localname);

        sleep(2);
}


/* 表示愿意接受文件 */
void transfer_file_y(int client_socket)
{
        Msg msg;
        msg.cmd = 10;
        strcpy(msg.fromname,myname);
        strcpy(msg.localname,mylocalname);
        strcpy(mylocalname,"\0");       //用完后置零
        write(client_socket, &msg, sizeof(Msg));
}

/* 表示不愿意接受文件 */
void transfer_file_n(int client_socket)
{
        Msg msg;
        msg.cmd = -10;
        strcpy(msg.fromname,myname);
        strcpy(msg.localname,mylocalname);
        strcpy(mylocalname,"\0");       //用完后置零
        write(client_socket, &msg, sizeof(Msg));
}

/* 传出文件来源开始传输文件 */
void start_transfer_file(int client_socket)
{
        Msg msg;
```

```
        msg.cmd = 11;
        strcpy(msg.fromname,myname);
        strcpy(msg.signname,fsignname);
        strcpy(msg.localname,mylocalname);

        int fd = open(msg.signname,O_RDONLY);
        if(fd == -1)
        {
                perror("open");
                printf("传输失败\n");
                return ;
        }

        int ret = 0;
        int i = 0;
        while(ret = read(fd,msg.msg,SIZE))
        {
                if(ret == -1)
                {
                        perror("read");
                        break;
                }
                if (ret == 0)
                {
                        break;
                }
                msg.num = ret;
                write(client_socket, &msg, sizeof(Msg));
                usleep(10000);          //这个睡眠时间减缓传输速度,降低不可预祝错误出现
        }
        printf("文件复制完成传输\n");
        strcpy(fsignname,"\0");
        strcpy(mylocalname,"\0");

        close(fd);
}

/* 接受文件 */
void save_transfer_file(Msg * buf)
{
        int fd = open(buf->signname,O_WRONLY|O_CREAT|O_APPEND,0777);
        write(fd,buf->msg,buf->num);
        if(buf->num != SIZE)
        {
                printf ("文件接受完成\n");
```

```
        }
        close (fd);
}

int main()
{
        int client_socket = Connect_Server();
        Ask_server(client_socket);
        close(client_socket);
        return 0;
}
```