网络安全技术课程实验报告

**实验一**



**实验名称：\_\_\_\_基于DES加密的TCP聊天程序\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_王泽豪\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**学号：\_\_\_\_\_\_\_\_\_1911483\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**专业：\_\_\_\_\_\_计科\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**提交日期：\_\_\_\_2022-3-18\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## 一、实验目的

①理解DES加解密原理。

②理解TCP协议的工作原理。

③掌握linux下基于socket的编程方法。

## 二、实验要求及要点

①利用socket编写一个TCP聊天程序。

②通信内容经过DES加密与解密。

## 三、实验内容

# 1.具体过程

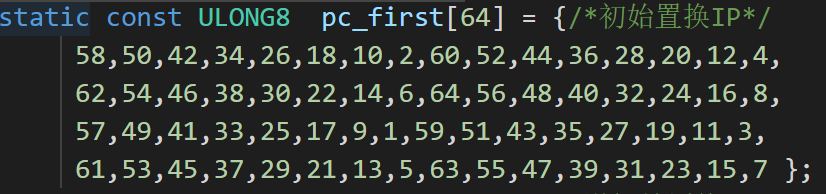
### DES算法细节

DES明文分组长度为64bit（不足64 bit的部分用0补齐），密文分组长度也是64bit。加密过程要经过16圈迭代。初始密钥长度为64 bit，但其中有8 bit奇偶校验位，因此有效密钥长度是56 bit，子密钥生成算法产生16个48 bit的子密钥，在16圈迭代中使用。解密与加密采用相同的算法，并且所使用的密钥也相同，只是各个子密钥的使用顺序不同。

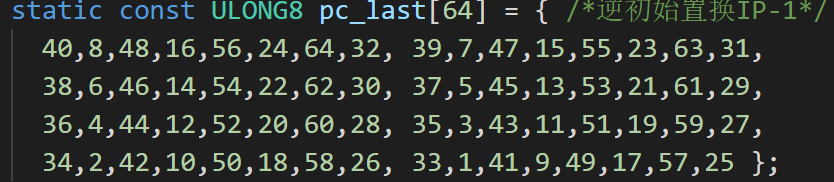
DES算法的全部细节都是公开的，其安全性完全依赖于密钥的保密。算法包括初始置换IP、逆初始置换IP-1、16轮迭代以及子密钥生成算法。

#### 初始置换IP

将64 bit的明文重新排列，而后分成左右两块，每块32bit，用L0和R0表示。IP置换表如表2-1所示。通过对该表进行观察可以发现其中相邻两列的元素位置号数相差8，前32个元素均为偶数号码，后32个均为奇数号码，这样的置换相当于将明文的各字节按列写出，各列经过偶采样和奇采样置换后，再对其进行逆序排列，将阵中元素按行读出以便构成置换的输出。



#### 逆初始置换IP-1

在16圈迭代之后，将左右两端合并为64bit，进行逆初始置换IP-1 ，得到输出的64bit密文。

#### 16轮迭代

16轮迭代是DES算法的核心部分。将经过IP置换后的数据分成32bit的左右两段，进行16圈迭代，每轮迭代只对右边的32bit进行一系列的加密变换，在一轮加密变换结束时，将左边的32bit与右边进行异或后得到的32bit，作为下一轮迭代时右边的段，并将这轮迭代中的右边段未经任何加密变换时的初始值直接作为下一轮迭代时左边的段，这需要在每轮迭代开始时，先将右边段保存一个副本，以便在该轮迭代结束时，将该副本直接赋值给下一轮迭代的左边段。在每轮迭代时，右边的数据段要经过的加密运算包括选择扩展运算E、密钥加运算、选择压缩运算S和置换P，这些变换合称f函数。

1. 选择扩展运算

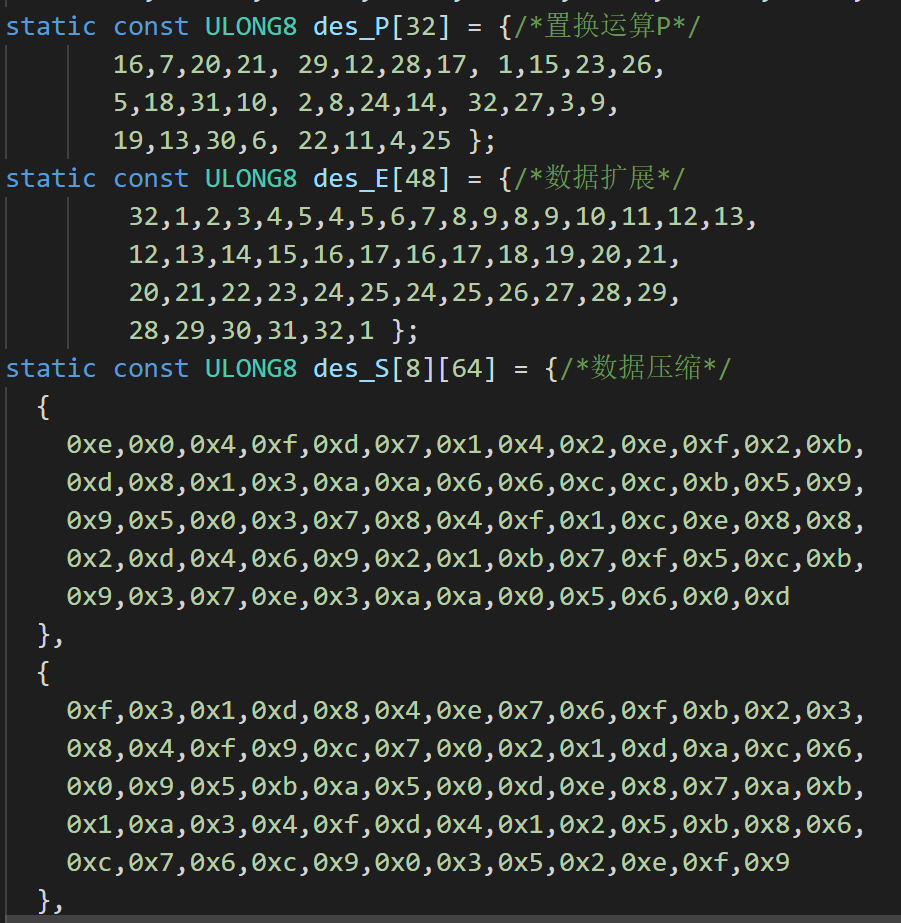
选择扩展运算（也称为E盒）的目的是将输入的右边32bit扩展成为48bit输出。置换结果按行输出的结果即为密钥加运算48bit的输入。

1. 密钥加运算

密钥加运算，是将选择扩展运算输出的48bit作为输入，与48bit的子密钥进行异或运算，异或结果作为选择压缩运算（S盒）的输入。

1. 选择压缩运算

选择压缩运算（S盒）是DES算法中唯一的非线性部分，它是一个查表运算，共有8张非线性的变换表，如表2-4至2-11所示，每张表的输入为6bit，输出为4bit。在查表之前，将密钥加运算的输出作为48bit输入，将其分为8组，每组6bit，分别进入8个S盒进行运算，得出32bit的输出结果作为置换运算的输入。



（4）置换运算

置换运算P是一个32bit的换位运算，对选择压缩运算输出的32bit数据按表2-12进行换位。将数据R1R2R3…R31R32转换成为R16R7R20…R4R25。

#### 子密钥生成

64bit初始密钥经过置换选择PC-1、循环左移运算LS、置换选择PC-2，产生16轮迭代所用到的子密钥ki。初始密钥的第8、16、24、32、40、48、56、64位是奇偶校验位，其余56位为有效位。下面以第N轮子密钥的产生为例进行说明。

2．置换选择PC-1

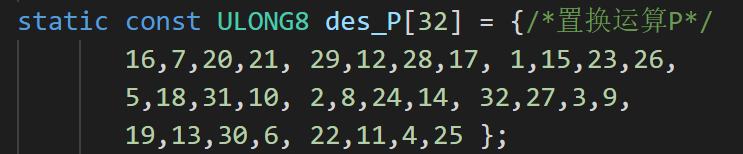
置换选择PC-1只在第一轮子密钥的产生过程中需要使用，它的目的是从64bit初始密钥中选出56bit有效位。选择的过程是一个查表的过程，如表2-13与2-14所示，输出的56bit被分为两组，每组28bit，分别进入C寄存器（查表2-13的结果）和D寄存器（查表2-14的结果）中，准备进行循环左移。

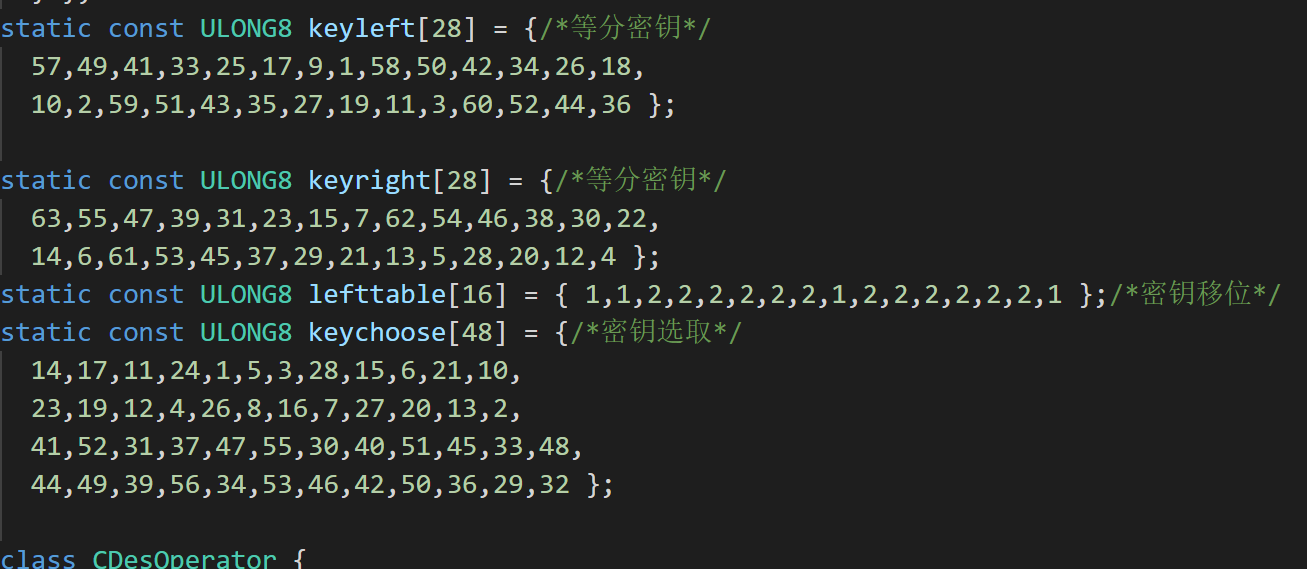
3．循环左移LS：

此轮密钥的产生所需要循环左移的位数即为表2-15中的第N个元素（首元素为第一个）。C，D寄存器中的28bit经过循环左移后，拼接为56bit作为此轮置换选择PC-2的输入，同时也作为第N+1轮子密钥循环左移的输入。

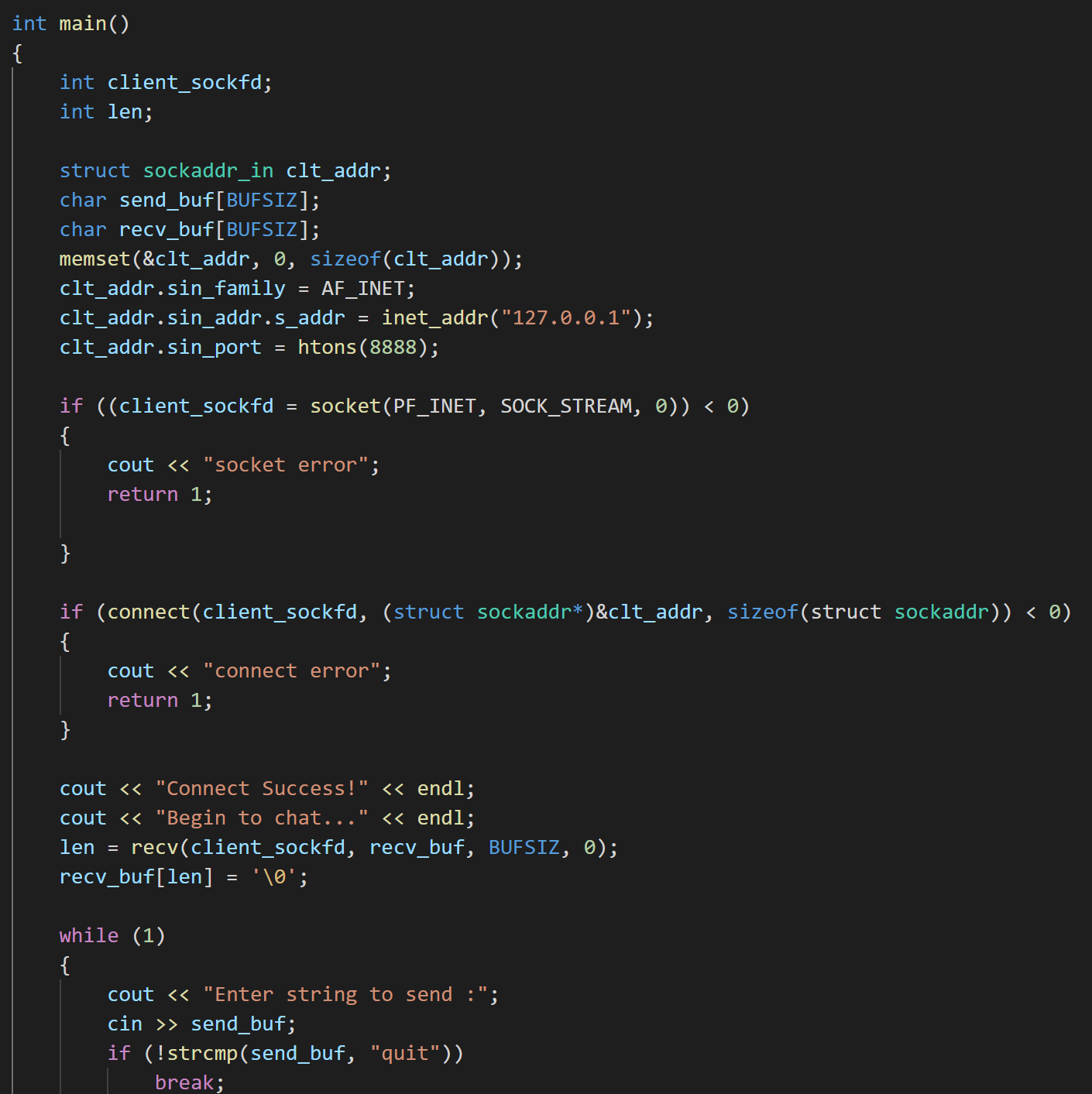
4.置换选择PC-2

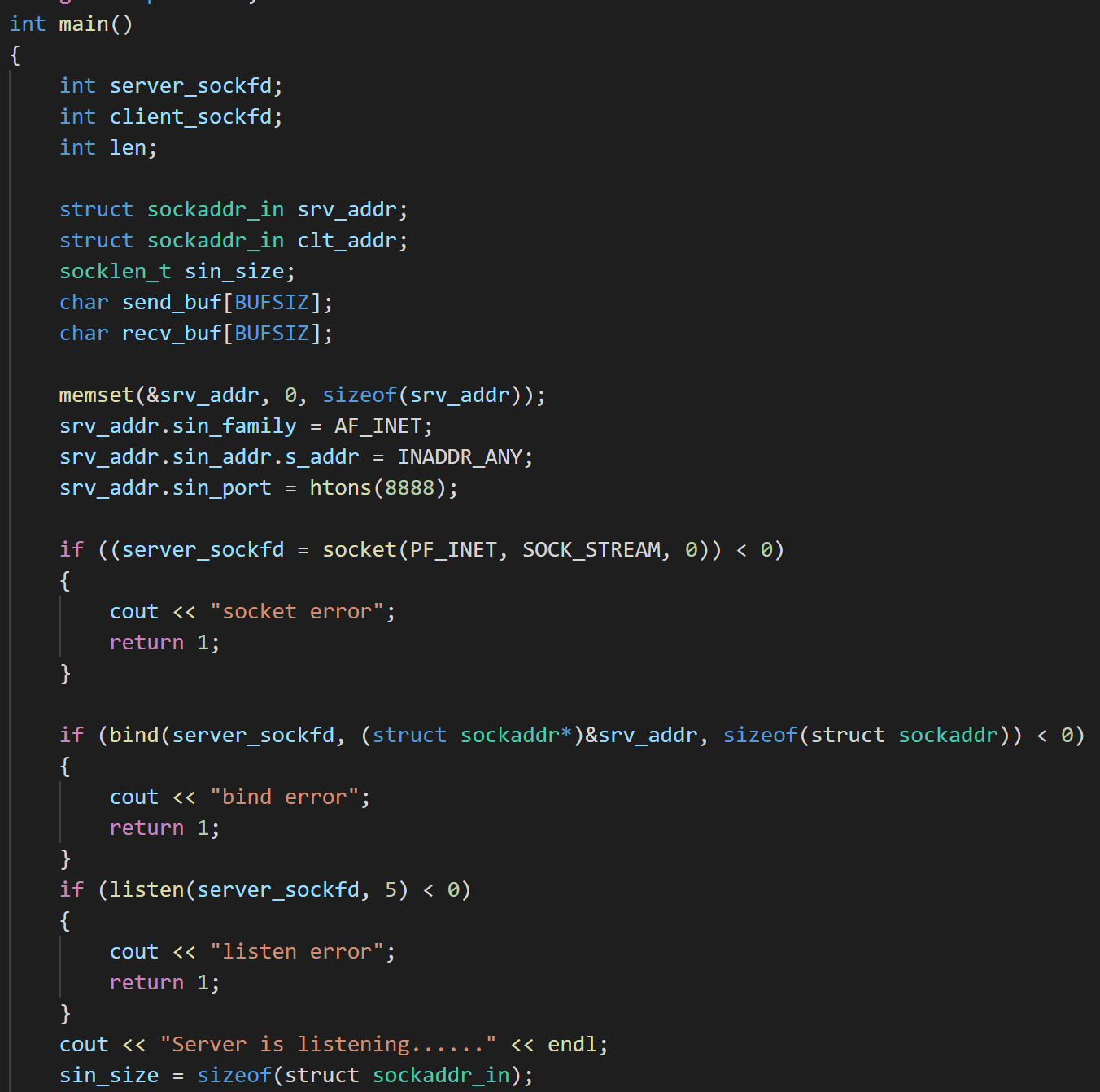
置换选择PC-2将输入的56bit中的第9、18、22、25、35、38、43、54位删去，将其余位置按照表2-16置换位置，输出48bit，作为第N轮的子密钥。



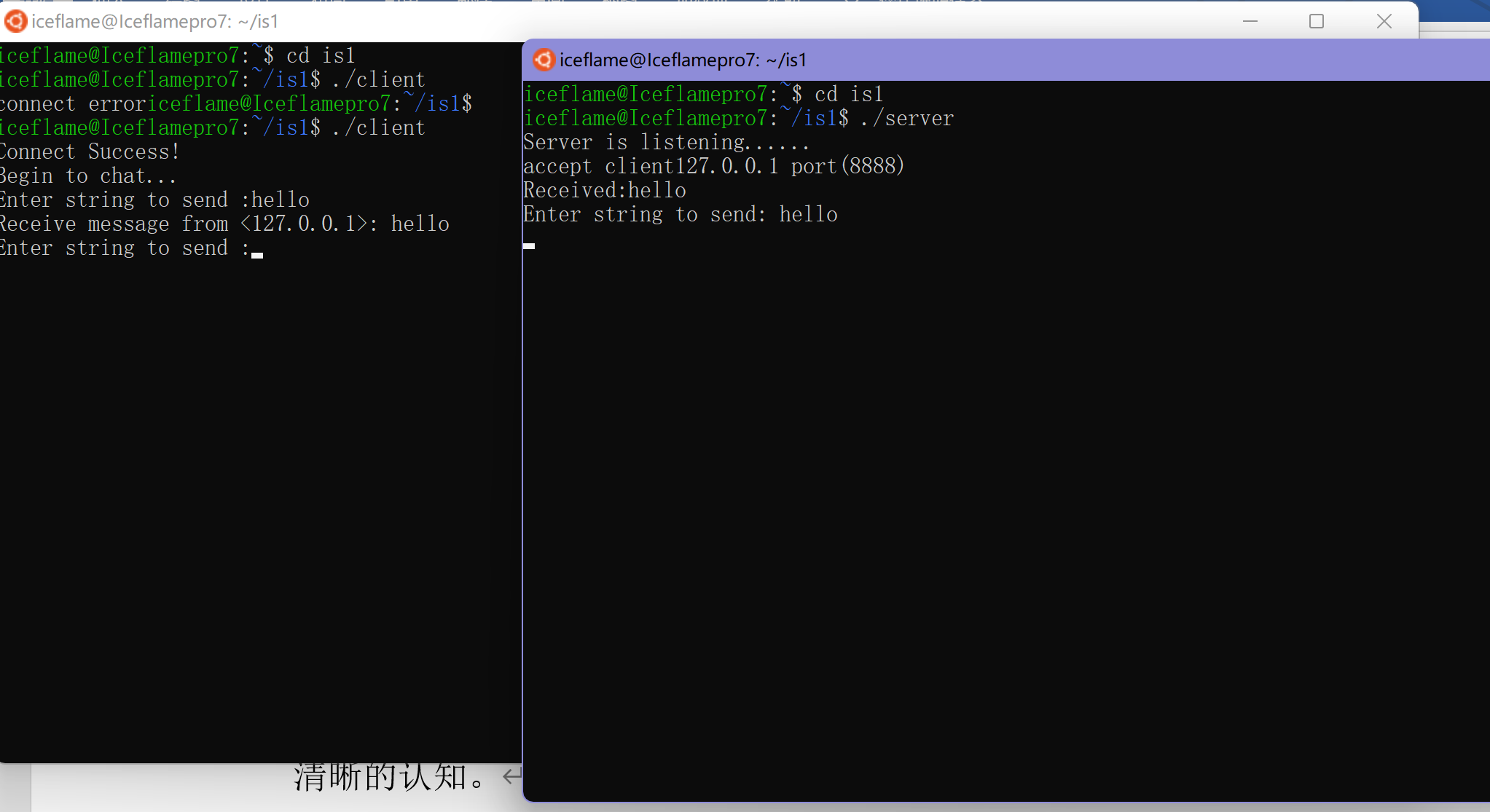


**利用套接字建立服务端与客户端之间的TCP连接**：





# 2.结果截图



# 3.遇到的问题与解决方法

1. 实验中使用了kbhit()函数，以用来检测键盘是否输入；此头文件在windows平台的C编译器中被提供，但在linux平台中没有，故手动写了头文件”socket.h”。
2. 由于TCP在某些特殊情况下，recv()函数一次不能返回对方发送的全部数据，因而在父进程中从socket接收数据使用了TotalRecv()函数

## 四、个人总结

1.通过这次实验我更加细致地了解了TCP连接的建立过程；

2.学习了通过linux创建简单的多进程的方法；

3.更深层的了解了套接字的作用以及使用方法，对网络连接的过程有了更加清晰的认知。

4.较为全面地了解了DES加密的各个阶段的具体内容。