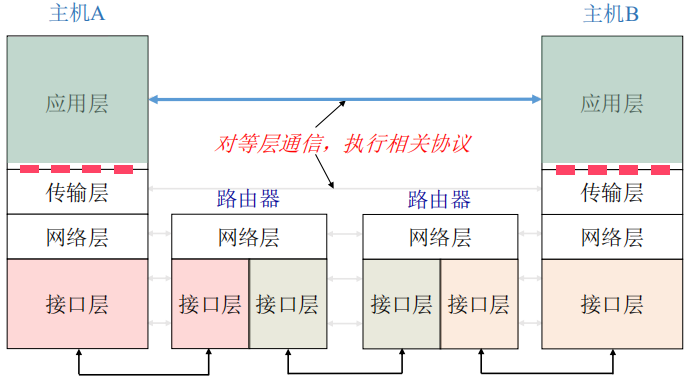
**实验一 使用套接字实现聊天室**

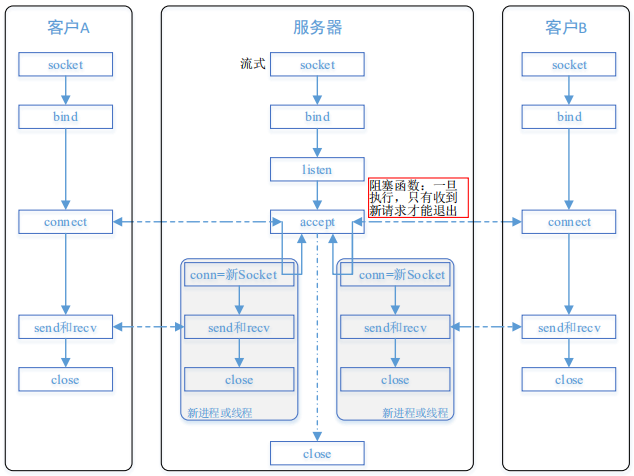
1813069 郭怡霏

1. 实验目的：
2. 给出你聊天协议的完整说明。
3. 利用C或C++语言，使用基本的Socket 函数完成程序。不允许使用CSocket等封装后的类编写程序。
4. 使用流式Socket完成程序。
5. 程序应有基本的对话界面，但可以不是图形界面。程序应有正常的退出方式。
6. 完成的程序至少应能实现两个用户之间的英文和中文聊天。
7. 编写的程序应结构清晰，具有较好的可读性。
8. 实验原理：



如上图所示，根据TCP/IP的体系结构，主机自顶向下分为应用层、传输层、网络层、接口层四层，当主机A向主机B发送信息时，信息由A自顶向下层层封装，经由路由器传递至B的底部接口层之后自底向上层层解封，B的应用层获得数据。A与B的信息通过以上的途径传输，但两台主机对等层的信息的识别与交流需执行相关的协议。进程是主机中运行的程序，不同主机上的进程通信通过交换信息来完成。本次实验中，采用较为常见的客户/服务器（C/S）模型，客户向服务器发出服务请求并接收服务器的响应，服务器等待客户的请求并为客户提供服务。应用层的协议包括信息的类型、语法、语义、处理方式等内容，需自行定义；传输层可提供TCP和UDP服务，TCP服务具有面向连接、可靠传输、流量控制、拥塞控制等特性，而UDP服务则不可靠，不能提供流量控制、拥塞控制等。

socket（套接字）是TCP/IP网络操作系统为网络程序开发提供的典型网络编程界面，分为数据报套接字和流式套接字。本次实验使用流式套接字，使用TCP协议，支持主机之间面向连接的、顺序的、可靠的、全双工字节流传输（通讯可双向同时传输）。流式套接字采用TCP服务，流程图如下图所示：



1. 实验重点：
2. 自定义应用层协议；
3. 使用基本socket函数，流式套接字；
4. 支持中英文聊天。
5. 实验环境：
6. 操作系统：Oracle VM VirtualBox 6.1.14 r140239 (Qt 5.6.2)

Ubuntu 18.04.5 LTS (64 bit)

1. IDE：Visual Studio Code 1.50.1
2. 使用语言：C
3. 使用库：stdio.h string.h stdlib.h unistd.h netinet/in.h arpa/inet.h sys/socket.h

pthread.h time.h

1. 实验步骤：

自定义协议：

1. 统一使用1207号服务端口；
2. 单次发送消息长度不得超过100个字；
3. 客户端只输入“exit”时断开连接。

本实验由服务器和客户端两个部分构成，下面分别对其进行介绍：

1. server.c：

本文件为服务器源文件，由以下几部分构成：

1. 预设部分：此处包含了九个头文件，用宏定义限定了聊天室的人数上限， 设置了服务器的socket、可连接的客户端空数组、主机ip地址、所使用 的端口号、socket地址（来自socket.h）、用户名长度限制、时间。回送地 址（127.x.x.x）是本机回送地址（Loopback Address），即主机IP堆栈内部 的IP地址，主要用于网络软件测试以及本地机进程间通信，无论什么程 序，一旦使用回送地址发送数据，协议软件立即返回，不进行任何网络传 输；127.0.0.1分配给loopback 接口，loopback 是一个特殊的网络接口(可 理解成虚拟网卡)，用于本机中各个应用之间的网络交互。只要操作系统 的网络组件是正常的，loopback 就能工作。
2. init函数：创建了服务器的流式套接字serverSocket（来自socket.h），

更新并存储地址信息（来自in.h），完成socket和地址的bind操作（来 自socket.h），开始监听客户端的socket连接请求情况（来自socket.h）。

1. sendAll函数：在客户端连接数组中进行遍历，如果有连接的客户端，就

将发出操作写入监听日志，并将参数msg用send函数发送给该客户端（来

自socket.h）。

1. server\_thread：被用作线程的参数。实时检查，需要退出时，从客户端连

接数组中找到该客户端并清除连接，在监听日志中记录退出信息，终止线

程；否则把服务器接收到的信息进行发送（来自socket2.h、pthread.h）。

1. server函数：提示已经启动服务器，在一个大循环中不断地设置套接字地

址存放（来自in.h），使用accept函数接收来自客户端的连接请求（来

自socket.h），生成一个对应的小socket，分配一个线程并启动（来自

pthread.h）。

1. main函数：调用init函数和server函数。
2. client.c：

本文件为客户端源文件，由以下几部分构成：

1. 预设部分：同1（1）。
2. init函数：创建了客户端的流式套接字clientSocket（来自socket.h），更新 并存储地址信息（来自in.h），向客户端发起connect连接请求（来自 socket.h）。
3. recv\_thread：被用作线程的参数。实时接收从服务器发来的信息并显示。
4. start函数：创建一个用于接收消息的线程，send该用户进入聊天室的消息，

持续接收输入并发送至服务器，直到用户输入“exit”四个字母，最后发

送一条退出聊天室的消息（来自socket.h）。

1. main函数：调用init函数以初始化套接字，提示用户输入用户名并规范 化，进入聊天室，调用start函数，退出聊天室时以文字提示。

编译：

在vscode命令行打开当前文件夹，输入如下代码：



运行：

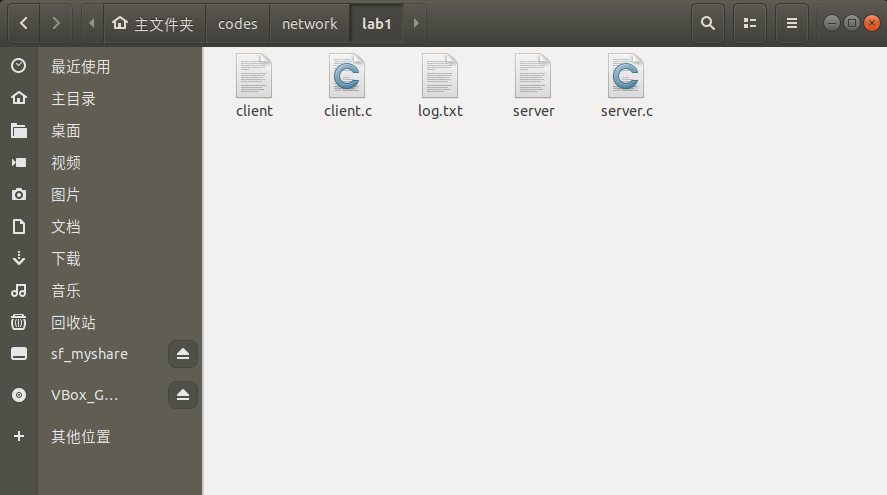
在命令行打开当前文件夹，输入如下代码，即可运行：



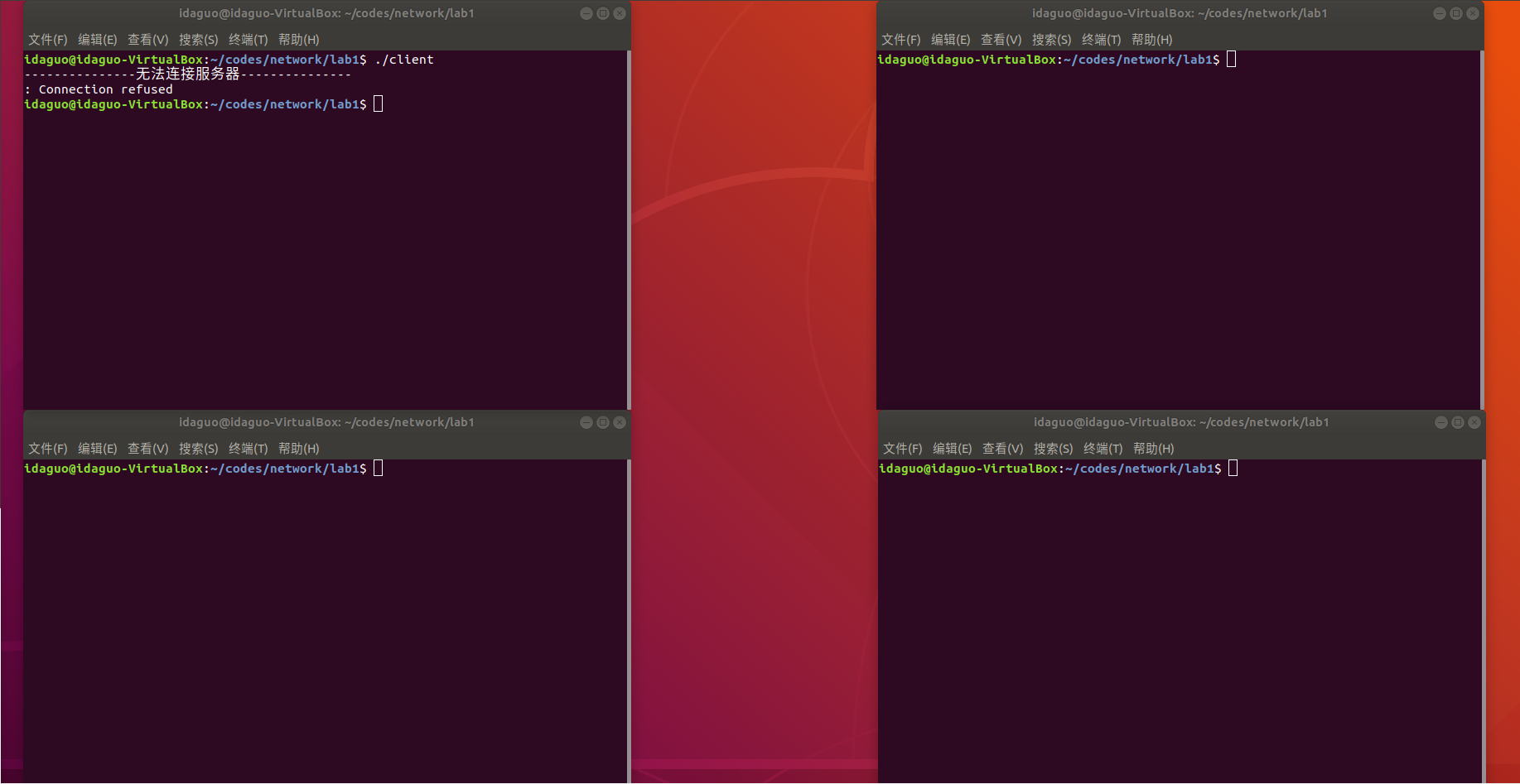


1. 实验结果：

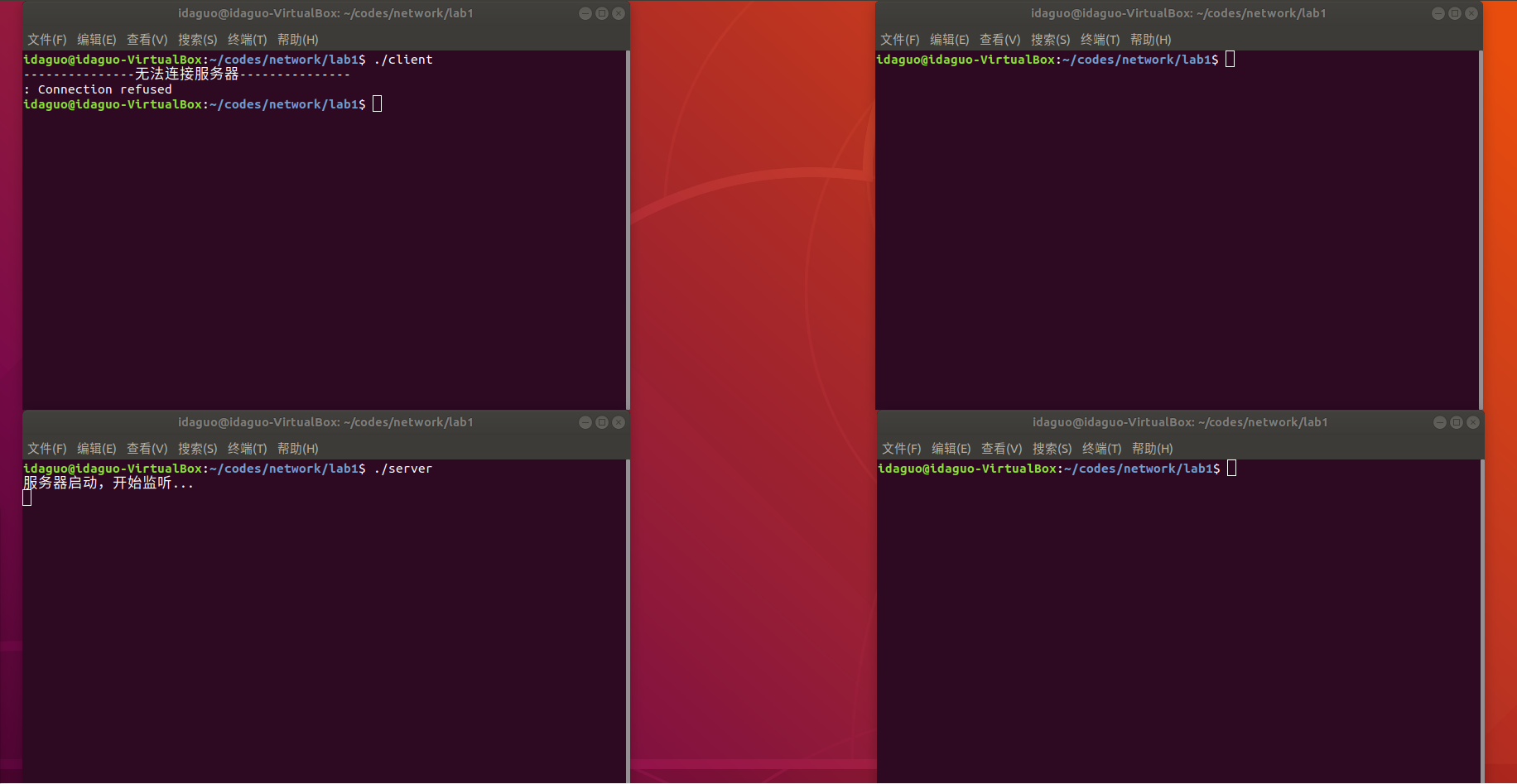
因可以设置聊天室人数上限，故为了便于演示，设置上限为2，即最多只能有两人同时在聊天室，演示如下：



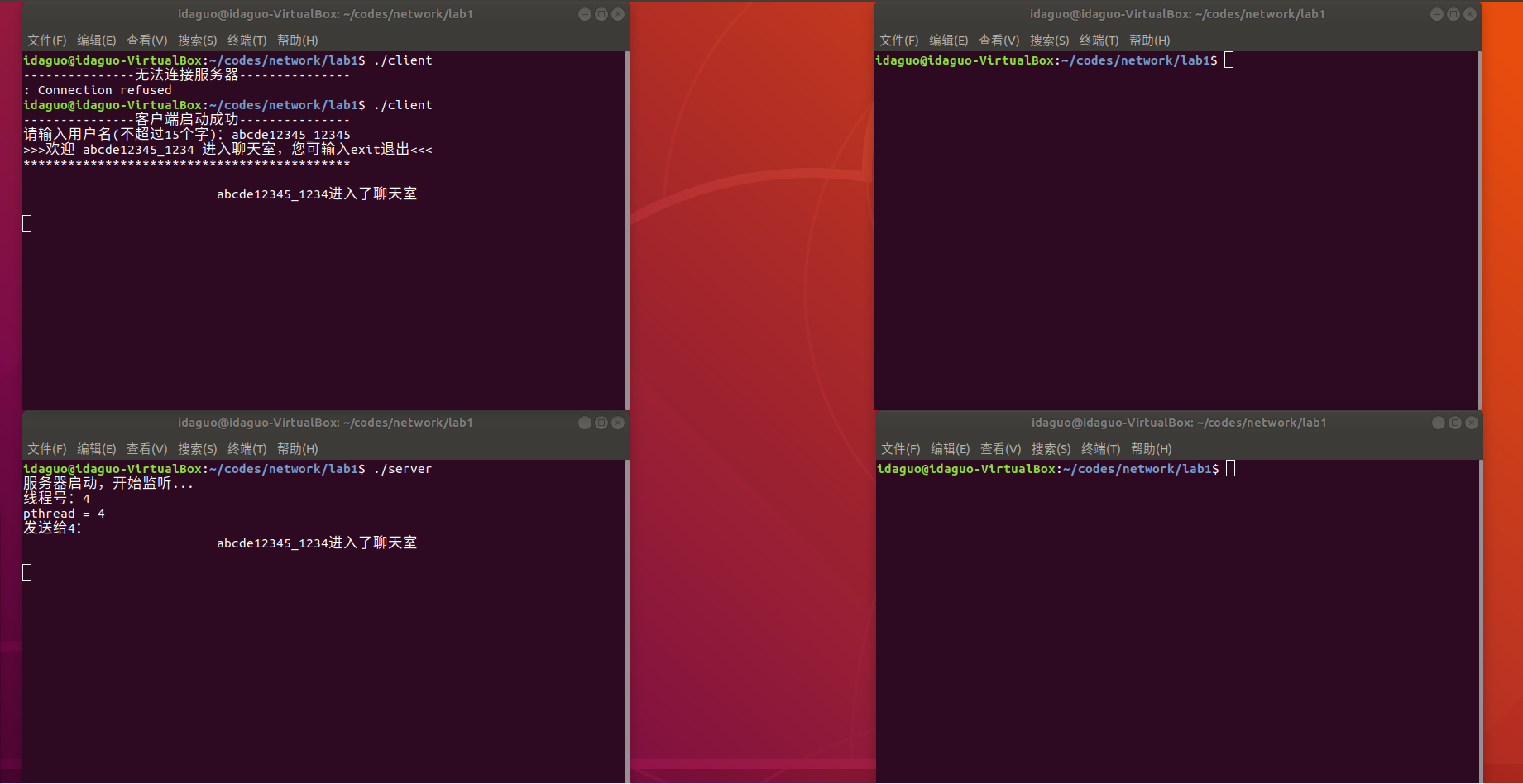
以上是文件列表，分别是客户端可执行文件、客户端源代码、监听日志、服务器可执行文件、服务器源代码。通过在命令行输入./xxx（xxx为可执行文件名）来运行。



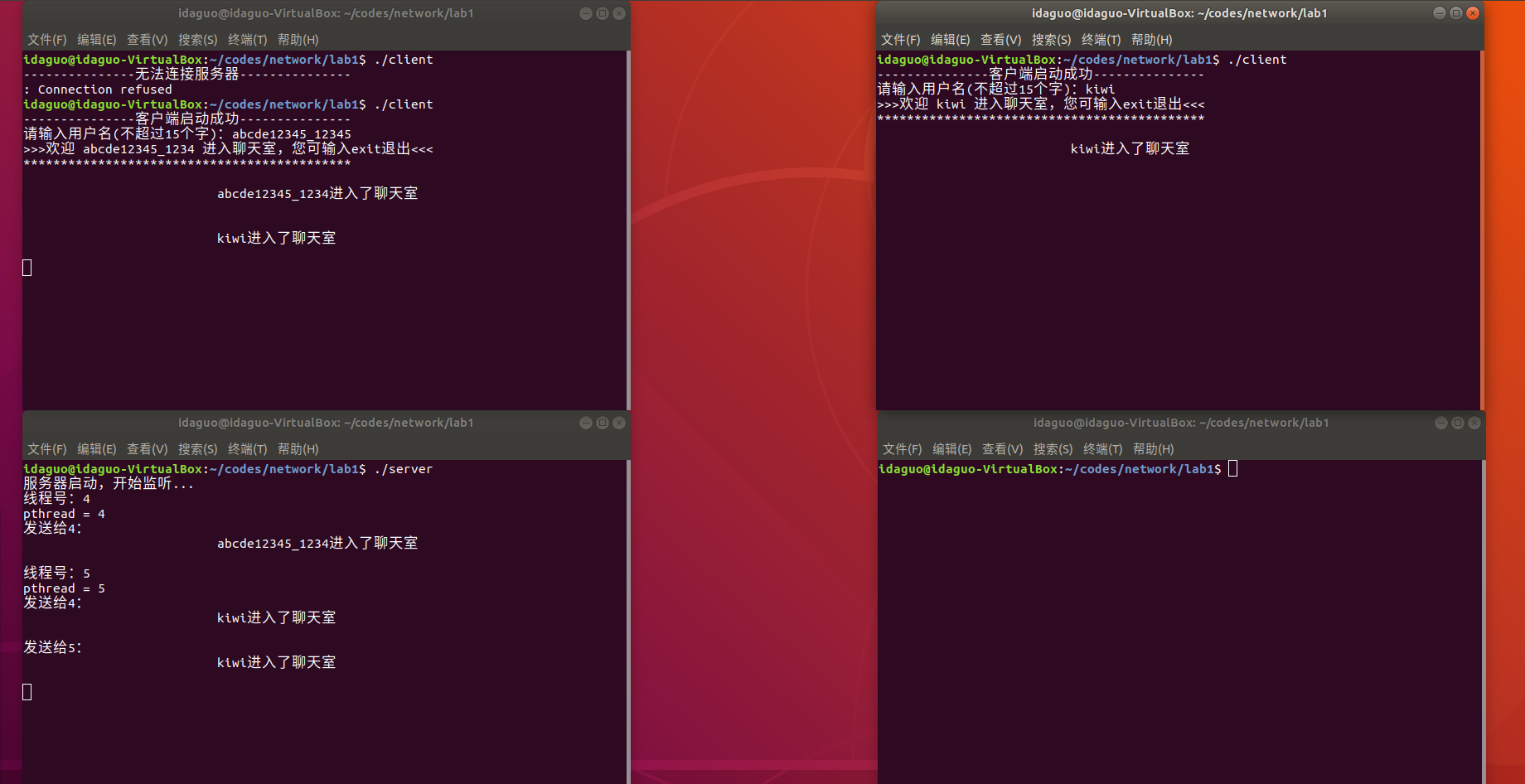
（1）在未开启服务器时开启客户端，显示无法连接



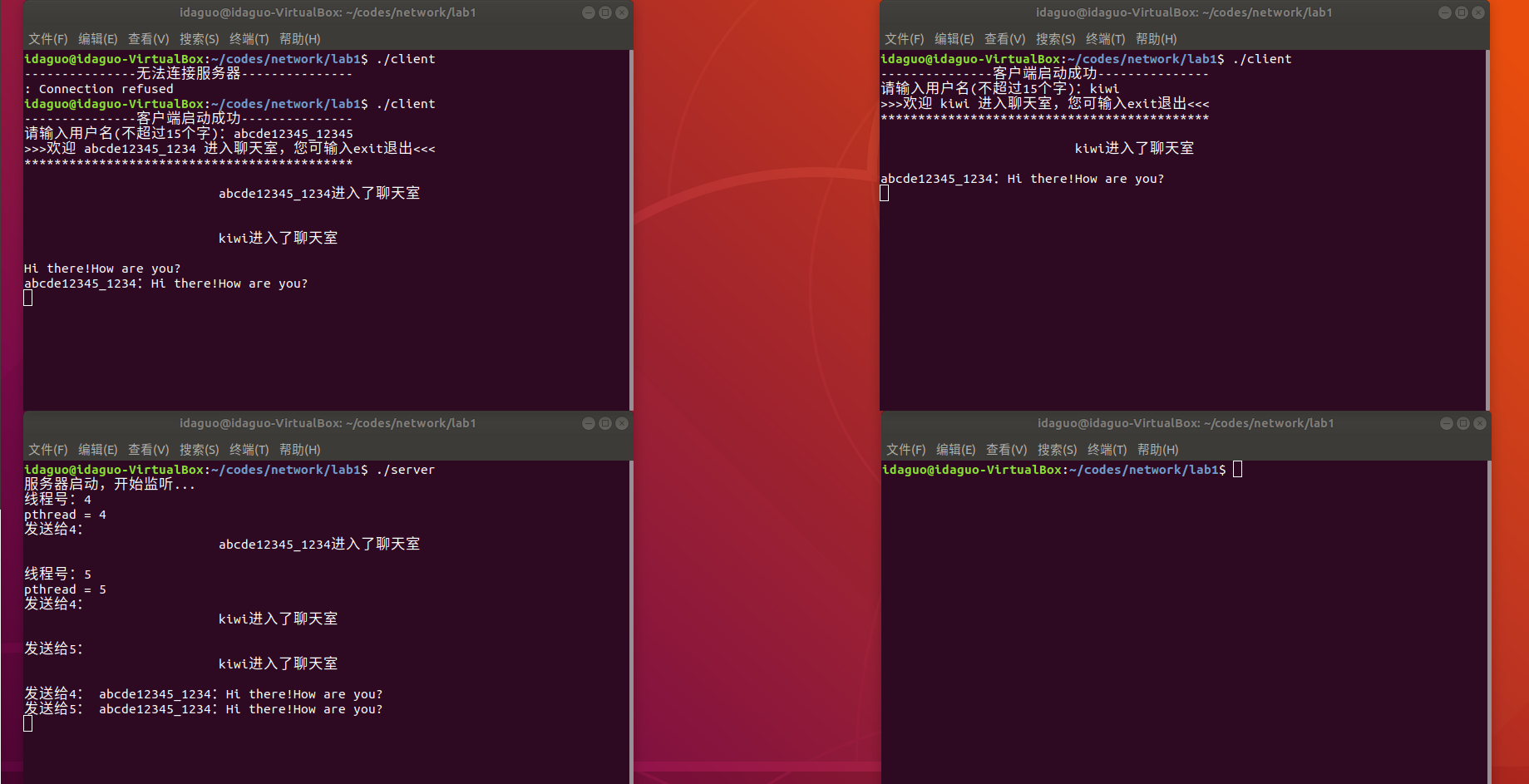
（2）开启服务器，服务器持续监听中



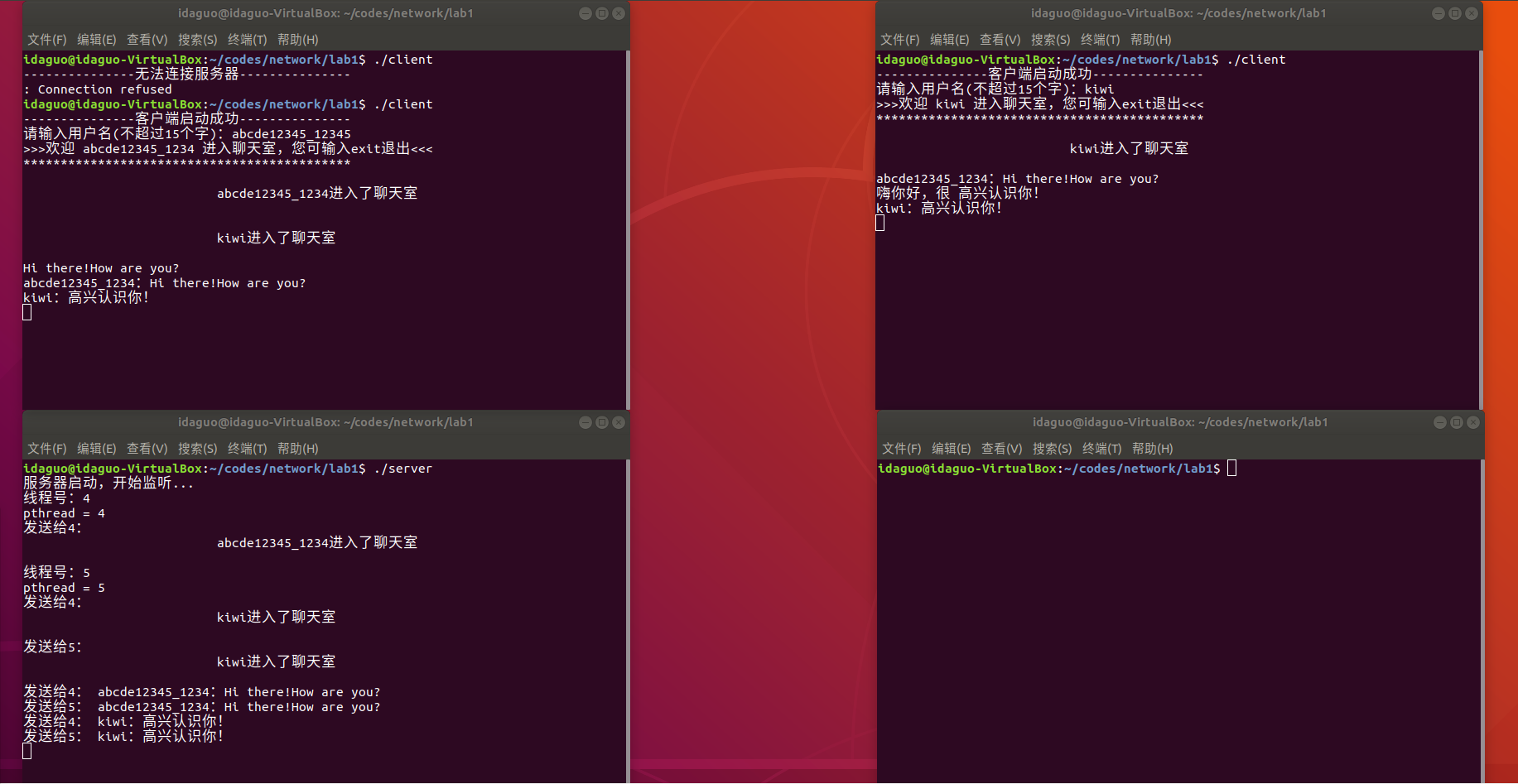
（3）开启客户端1，输入用户名abcde12345\_12345，只截取前15位，进入聊天室



（4）开启客户端2，输入用户名kiwi，进入聊天室

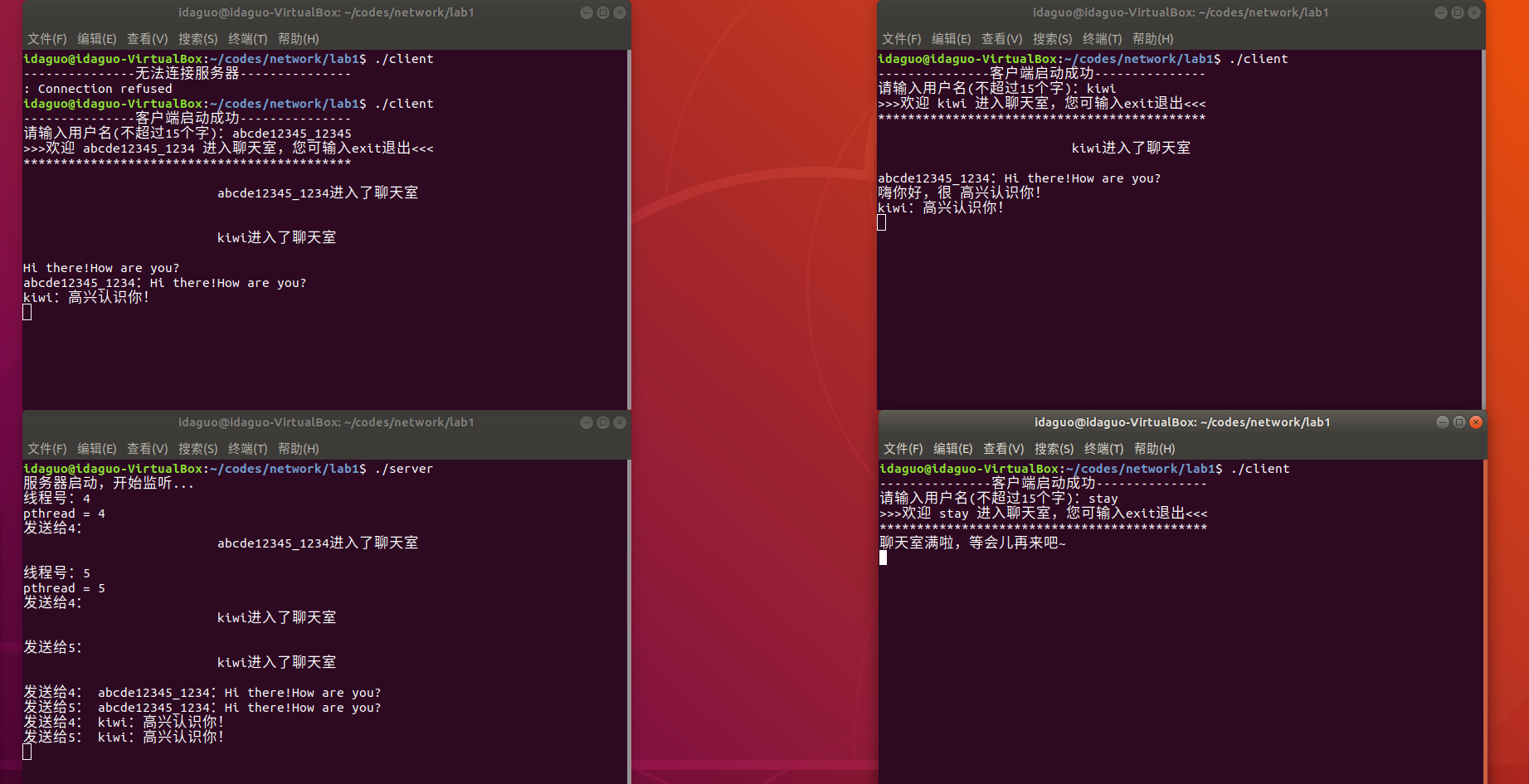


（5）用户1发送了英文信息，在用户1页面、用户2页面和服务器监听页面均有显示

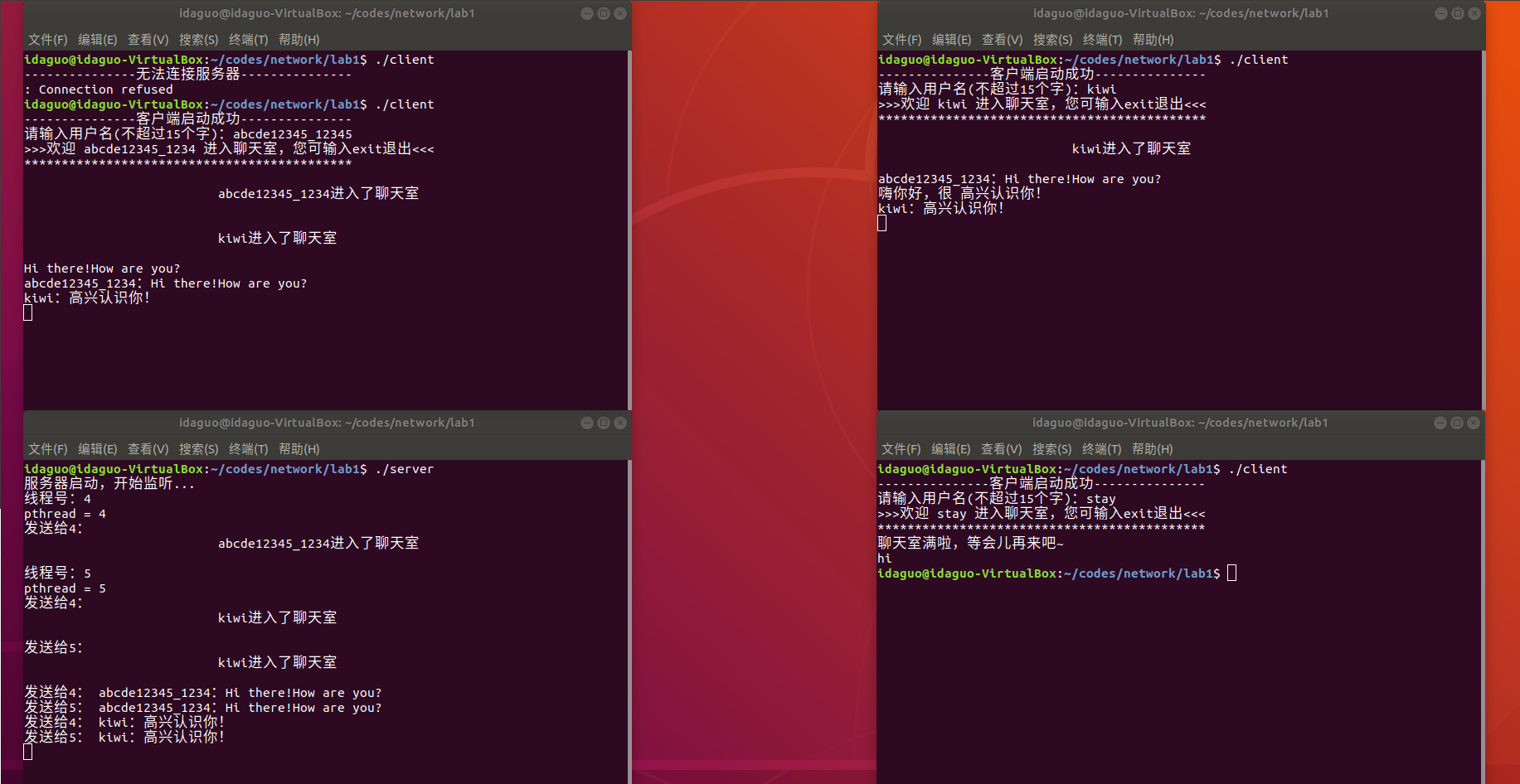


（6）用户2发送了中文信息，在用户1页面、用户2页面和服务器监听页面均有显示

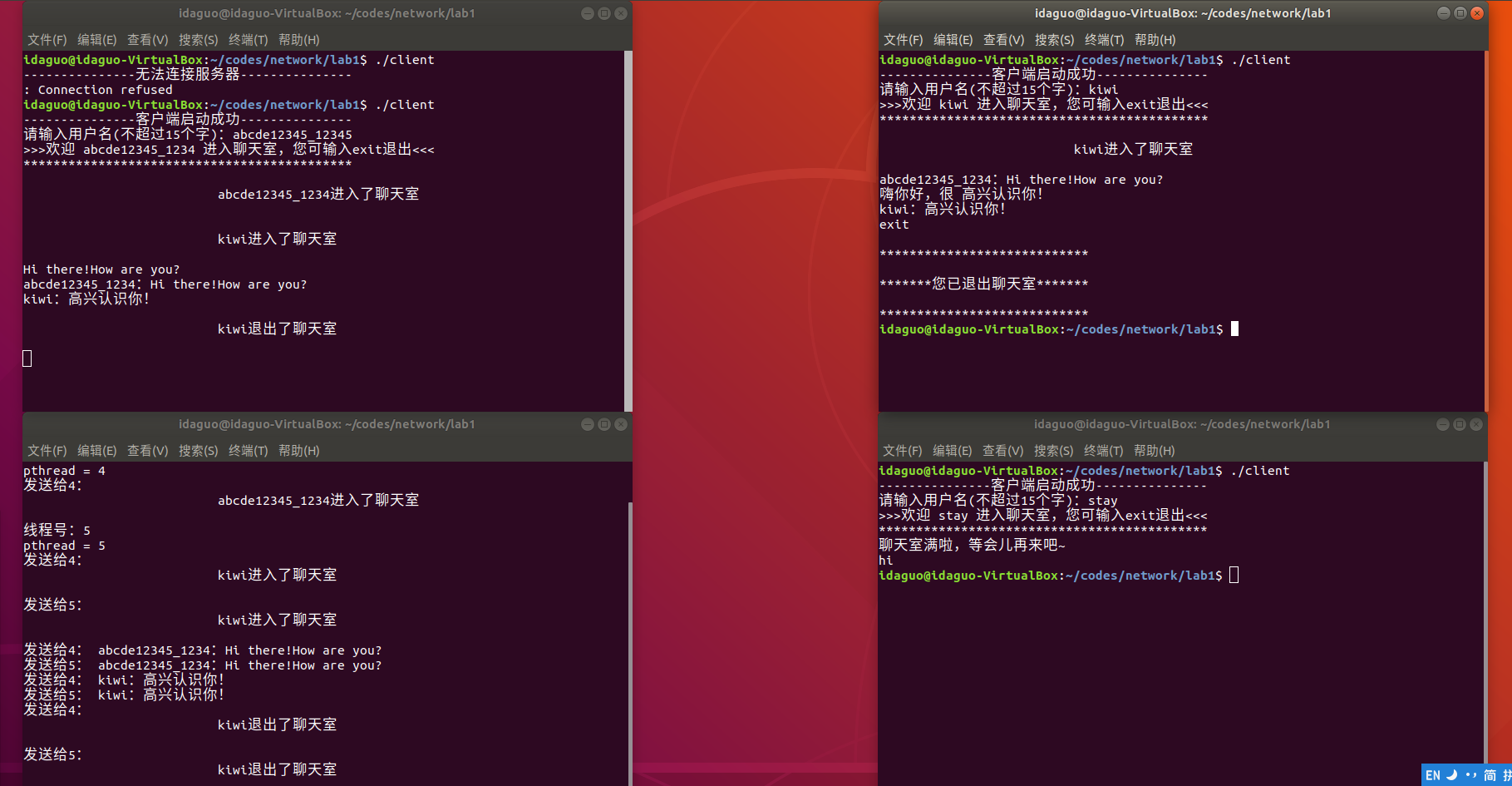
【此处输入中文之后按删除键，只能删除后一部分，前一部分只留在输入行，没有真正输入】



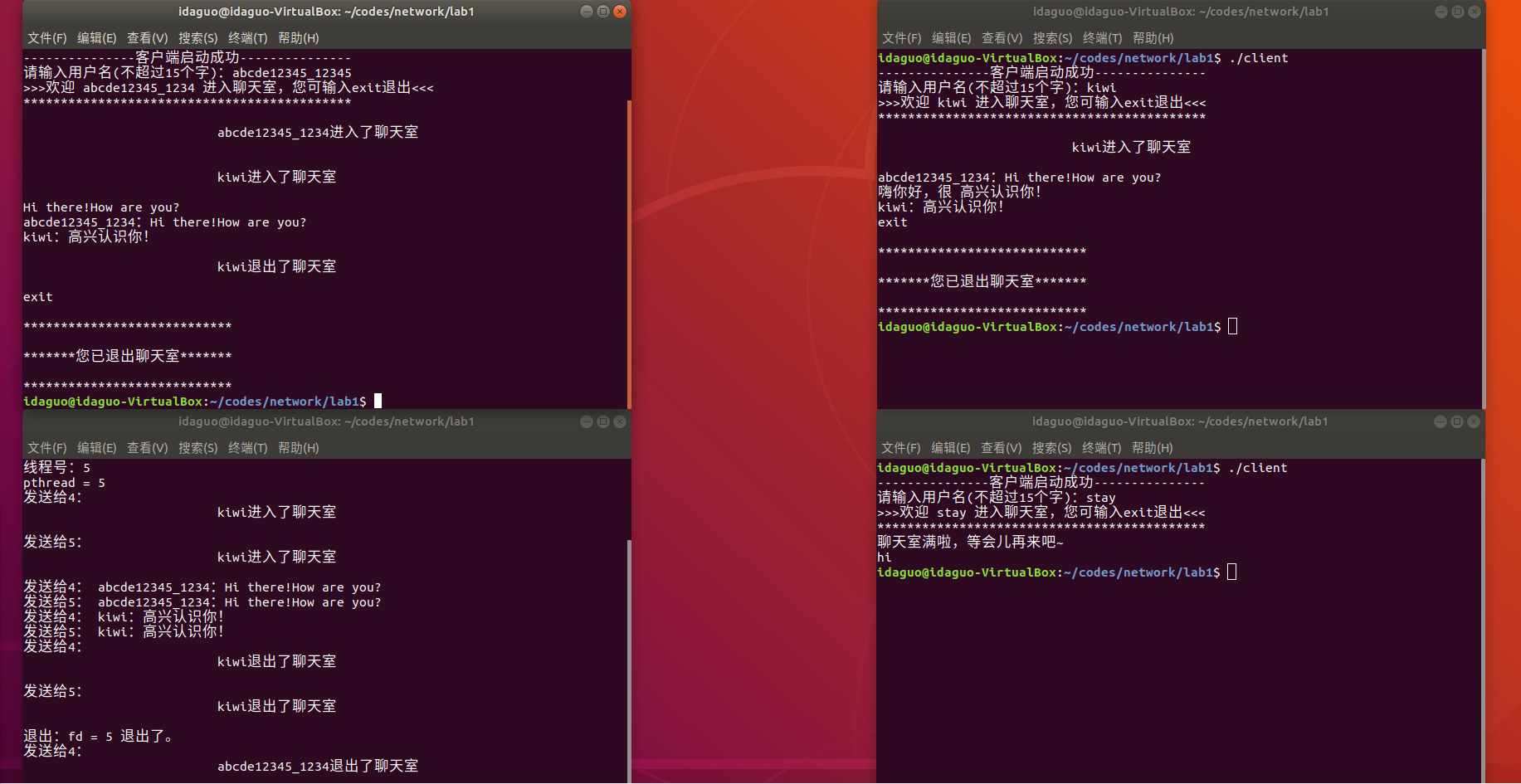
（7）开启客户端3，输入用户名stay，聊天室已满员，故无法进入聊天室



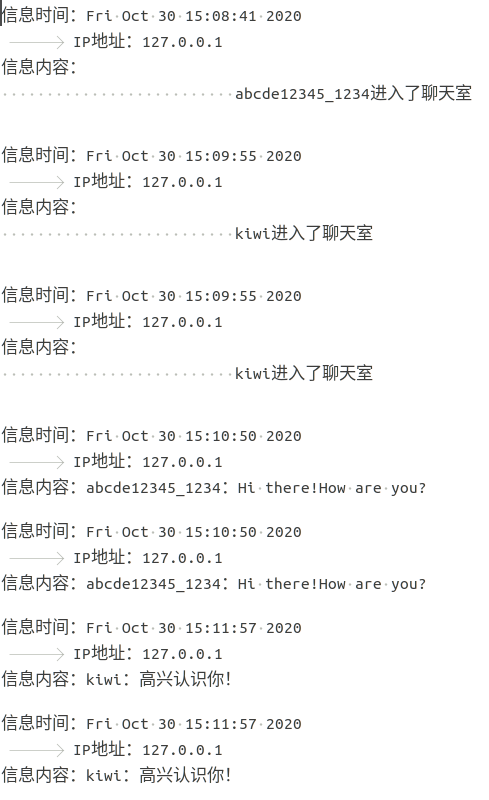
（8）客户端3不论输入什么都会直接退出，因为聊天室已满



（9）用户2退出聊天室，在用户1页面、用户2页面和服务器监听页面均有显示



（10）用户1退出聊天室，在用户1页面和服务器监听页面均有显示

（11）用gedit打开监听日志log.txt，如上图所示

1. 问题与不足：
2. 用户名输入，想限定在15个字以内，超过的全部截断，于是考虑了以下方法：
3. scanf(“%14s”,name)：超过的部分无法截断，留在后续的输出；
4. 前15个字用getchar，之后传入空循环：无法修改之前的昵称；
5. 先构造一个较长的缓冲区，输入之后再截取前15位送入name：若用户数过多，则空间消耗过大。

经过权衡，出于准确性和空间考虑，决定使用方法（2）。

1. 聊天输入，采用scanf("%s",inputBuf)的方式，发现输入英文时只读取了第一个单词，scanf函数无法读入空格，故设置限定100字输入缓冲区，采用逐个getchar()的方法来输入。经过检验，发现此处输入数字、英文字母、英文符号均可全部删除掉，但输入中文则删除不干净（见实验结果第6步），暂时没有想到如何解决。
2. 使用vscode的code runner插件直接编译运行，报错“对pthread\_create未定义的引用”：程序中使用了线程的操作，但pthread库不是Linux系统默认的库，连接时需要链接-lpthread，故在命令行使用gcc -o xxx xxx.c -lpthread命令生成可执行文件，再使用./xxx命令运行。