Linux大作业报告

**①题目选择：**

网络游戏——狼人杀

**②游戏流程及逻辑：**

狼人杀是一款多人在线网络游戏，本游戏支持6个人同时参与游戏，游戏角色分为狼人，平民，预言家，女巫；

🡪首先服务器等待6个玩家连接至服务器，每有一个玩家连接至服务器成功，服务器便随机为玩家分配身份，如果玩家在游戏开始之前断开连接，服务器会检测到玩家已断开连接并等待，直到6名玩家同时在线，游戏开始；在游戏开始前客户端输入的任何字符都不会发送到服务器；

🡪服务器向所有存活的玩家客户端发送游戏开始指令，游戏开始，首先天黑请闭眼，狼人开始行动，存活的狼人向服务器发送想要杀掉的玩家编号，如果某只狼人输入非法或者超时，它的输入将被忽略，如果两名狼人输入的目标不一致，系统将从两名狼人的目标中随机选择一名玩家kill；如果不是狼人的玩家在狼人行动期间输入，客户端将从标准输入中读取用户输入的字符，不做任何处理，因此并不会发送给客户端，直到服务器发来狼人行动结束的指令wolfover为止；

🡪女巫开始行动，女巫选择是否使用解药救被杀的玩家，女巫可以自救，是否使用毒药，如果使用毒药，输入使用的目标编号，一局游戏中女巫的每种药品只能使用一次；任何不合法或超时的输入服务器均会做出处理，令其无效；在女巫行动期间，任何非女巫玩家输入均无效，不会发送给服务器，直到服务器发送来女巫行动结束的指令womenover；

🡪预言家开始行动，预言家每晚可以查看一名玩家的身份，服务器向其返回所选择的玩家身份，A-平民 B-狼人 C-女巫 D-预言家，任何非法输入或超时，服务器将视为输入无效；在预言家行动期间，任何非预言家输入无效，不会发送到服务器端，直到服务器发来预言家行动结束的指令manover；

🡪服务器根据狼人，女巫和预言家的行动处理死亡玩家和发布结果，服务器断开死亡玩家的连接，并判断游戏是否结束；游戏结束标志：所有狼人都阵亡或者非狼人玩家全部阵亡；

🡪玩家讨论，存活的用户根据夜间的行动(如预言家验到了其他玩家的身份)发表自己的意见，

客户端首先先发送用户输入的数据到服务器，服务器收到数据后将其转发给其他仍存活的玩家；

🡪玩家讨论结束后，存活玩家将投票踢出一名玩家，用户的非法输入将视为弃权，如果多名玩家得票数相同，系统将随机踢出得票数相同的玩家中的一位；

🡪处理死亡玩家，如果上一步骤中踢出了一名有效玩家，服务器将断开该玩家的连接并判断游戏是否结束，若游戏结束，服务器将向存活到最后的玩家发送游戏结果，若游戏未结束，则进入新的一天，回到第二步循环执行，直到游戏结束为止。

**③所用知识及技术：**

Socket：多个客户可通过socket连接到服务器，6个人在线同时游戏，游戏的业务逻辑部分都在服务器端处理，而客户端主要用于接收用户的输入，发送到服务器，显示来自服务器的消息等；

异步阻塞IO：将socket设置为non-block，避免系统调用如accept时进程阻塞(进程或是线程执行到这些函数时必须等待某个事件的发生，如果事件没有发生，进程或线程就被阻塞，函数不能立即返回)，在非阻塞模式下如果循环使用select检测是否有连接请求或者是否有消息到达等(文件描述符集合发生变化)，将浪费CPU资源，使进程忙碌等待，因此我们设置了select的超时时间，以实现“非阻塞”；

同步和异步是针对应用程序和内核的交互而言的，同步指的是用户进程触发I/O操作并等待或者轮询的去查看I/O操作是否就绪，而异步是指用户进程触发I/O操作以后便开始做自己的事情，而当I/O操作已经完成的时候会得到I/O完成的通知，AIO通知有两种方式：线程回调函数和信号的方式。服务器端和客户端使用了AIO系统调用aio\_read和aio\_write函数。

阻塞和非阻塞是针对于进程在访问数据的时候，根据I/O操作的就绪状态来采取的不同方式，是一种读取或者写入操作函数的实现方式，阻塞方式下读取或者写入函数将一直等待，而非阻塞方式下，读取或者写入函数会立即返回一个状态值。

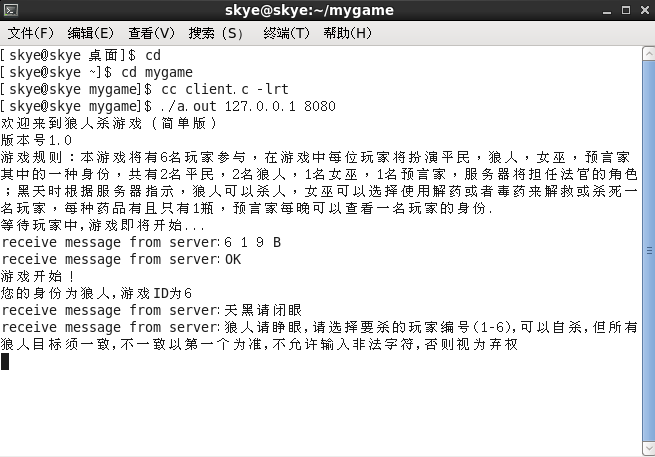
**④运行结果：**

服务器：cc server.c –lrt ./a.out 8080

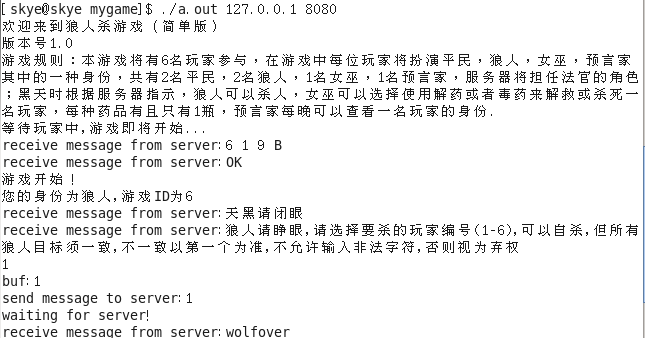
客户端：cc client.c –lrt ./a.out 127.0.0.1 8080

客户端连接服务器：(游戏已开始)

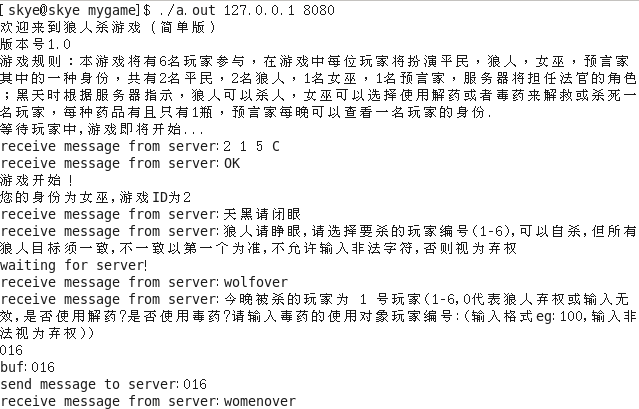
本局游戏中狼人为3 6号玩家，平民为1 5号玩家，女巫为2号玩家，预言家为4号玩家：



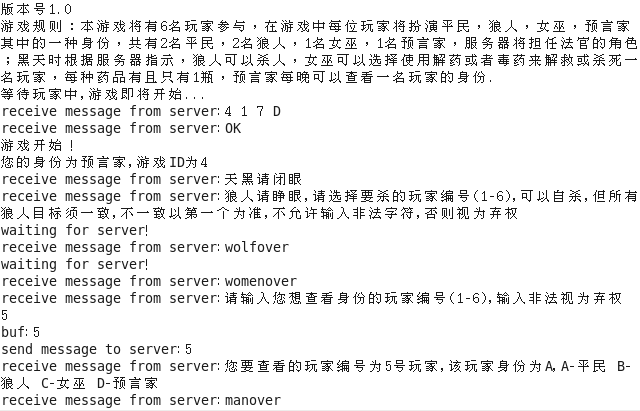
狼人选择杀掉1号玩家：



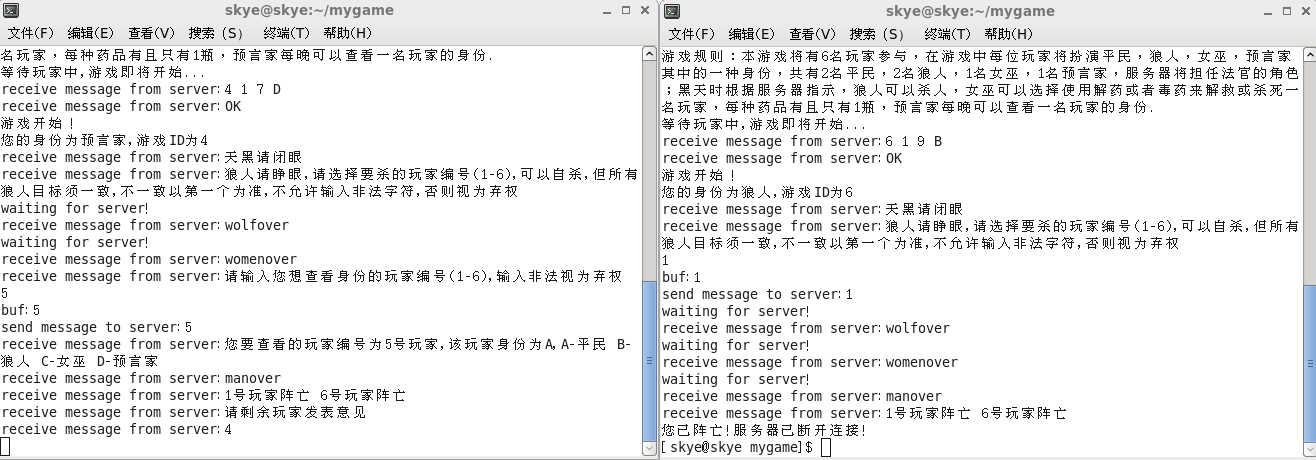
女巫可以看到昨晚1号玩家被杀，女巫输入016(是否使用解药/是否使用毒药/毒药的使用目标)，表示不使用解药，使用毒药杀掉6号玩家，女巫的合理输入为000,011,012,013,014,015,016,100,111,112,113,114,115,116，输入其他情况时，系统会根据情况进行处理，如006，017则仍不使用解药和毒药，在一局游戏中，女巫每种药品只能使用1次：



预言家选择要查看的玩家的编号：



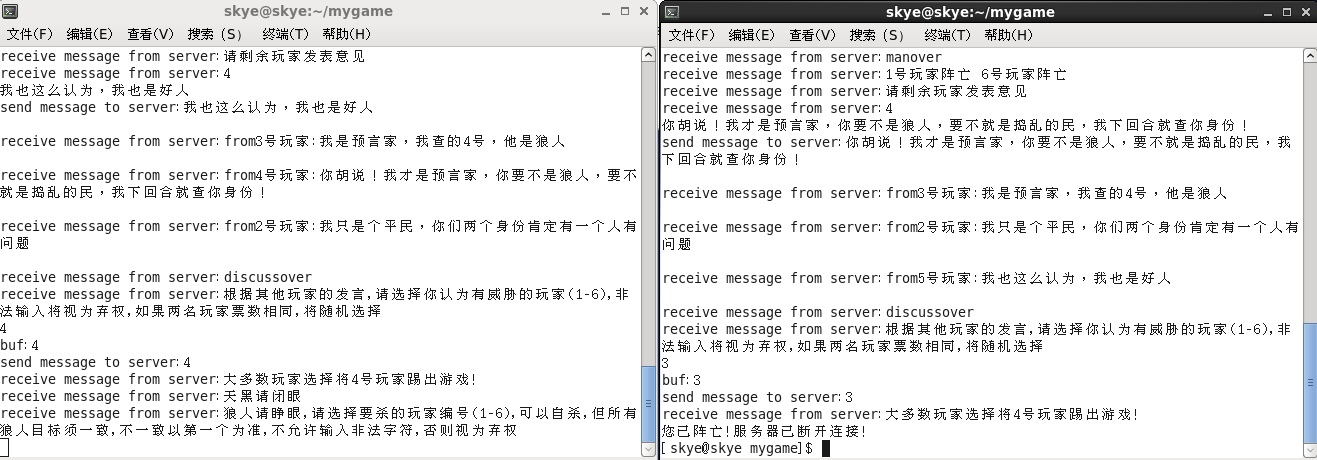
服务器处理死亡玩家并向所有玩家返回结果，死亡的玩家会断开连接：



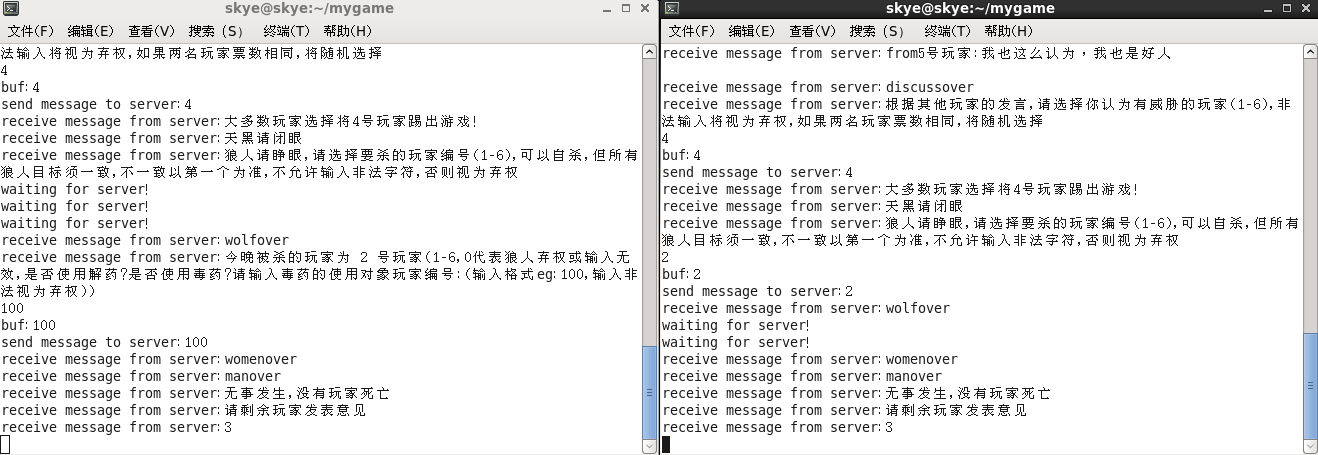
余下的存活的玩家可以相互交流，每位玩家只能发言一次：



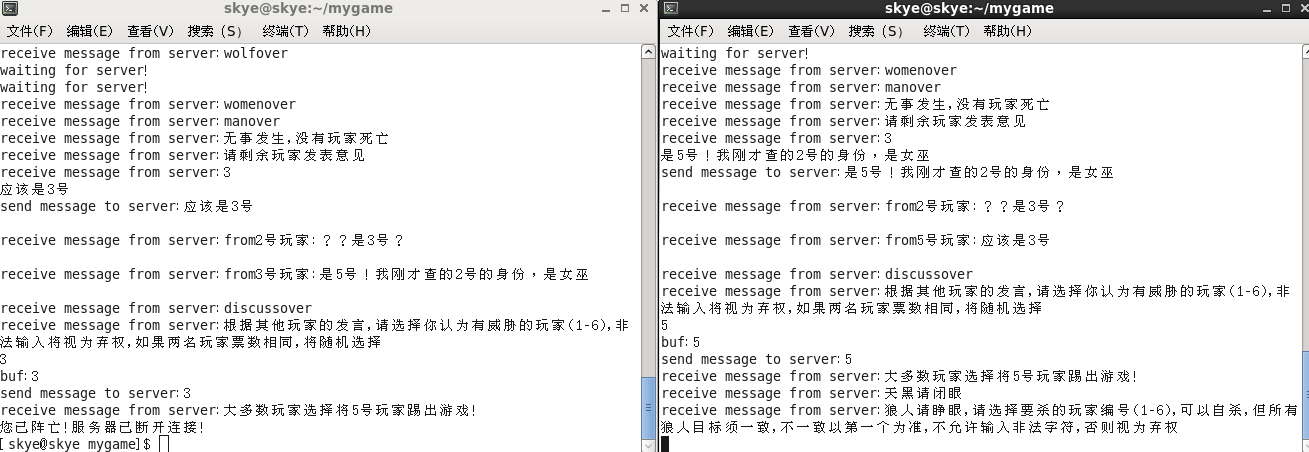
玩家投票后服务器处理，返回结果，并判断游戏是否结束，若未结束，则存活玩家继续游戏，进入第二个夜晚，业务逻辑同第一天夜晚：



狼人选择杀2号玩家，女巫发现被杀的是自己，选择使用解药自救，由于毒药已使用过，因此输入100，如果女巫输入11X(X∈【1,6】)，则服务器会自动处理，仍视为100：



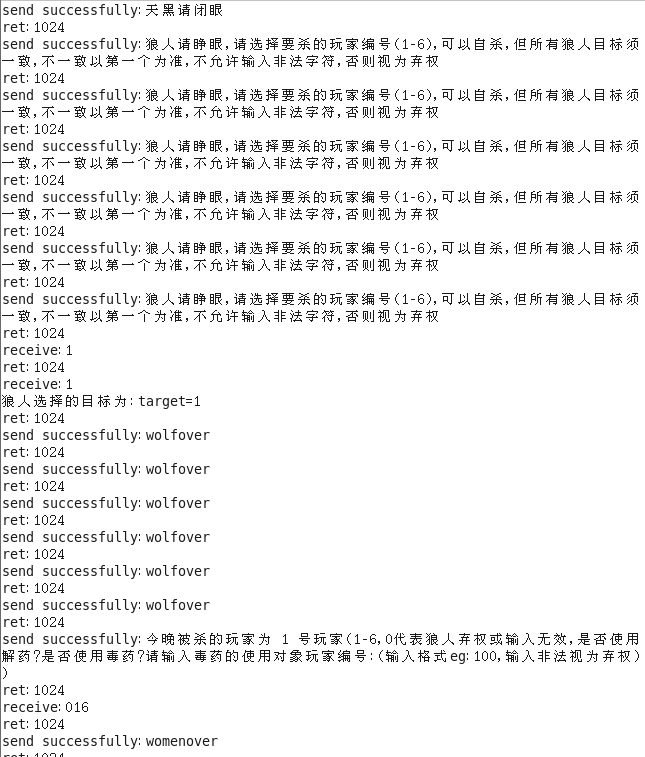
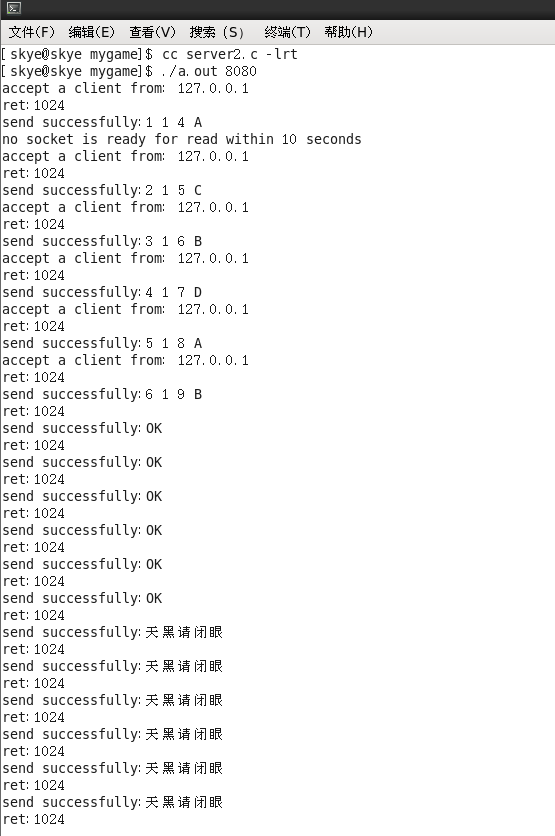
狼人发现杀的2号并没有死，怀疑其为女巫身份，因此发言时诱导平民及女巫，说出2号身份，若正确则坐实自己预言家的身份，投票后5号玩家阵亡：



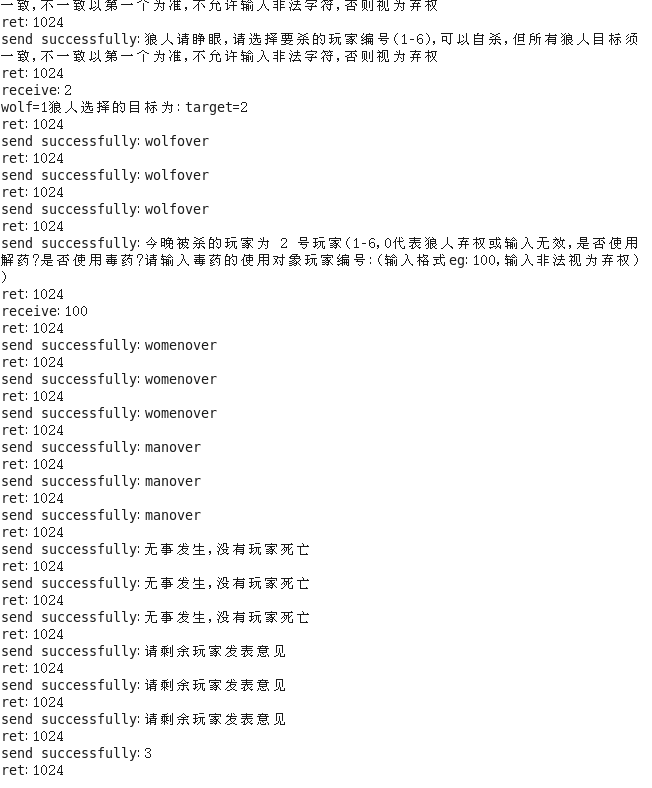
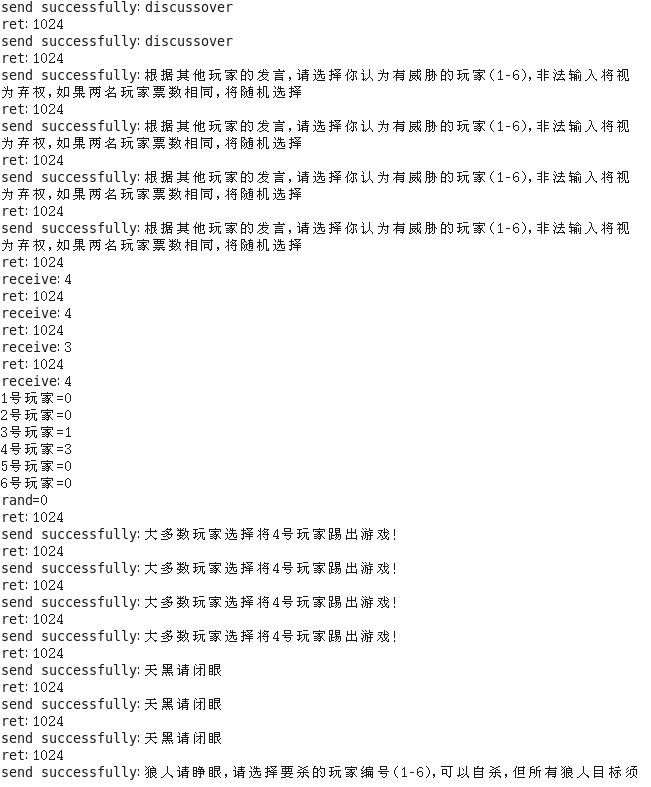
游戏继续，进入第三个夜晚，此时游戏已经结束，狼人只需杀掉2号女巫即胜利，游戏结束：

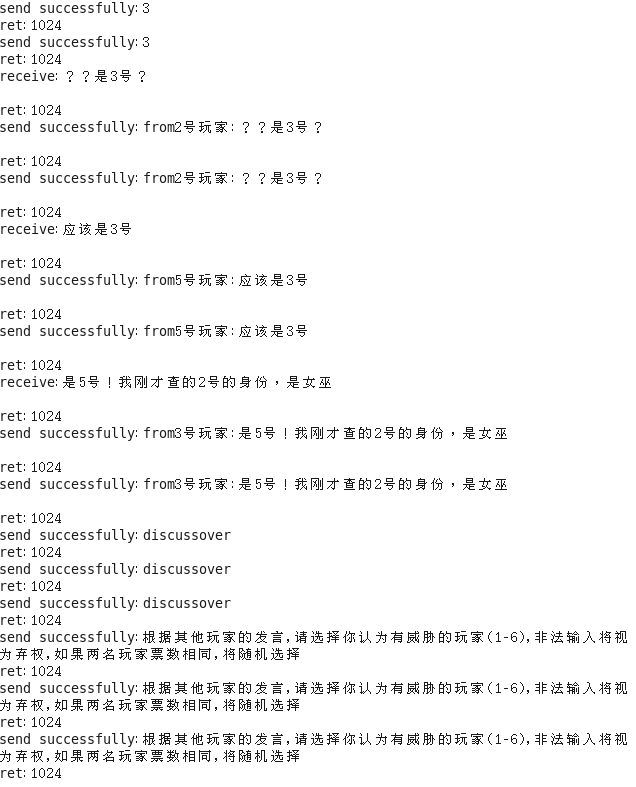


服务器端：









**⑤改进之处：**

该程序还有很多改进之处，如目前服务器只能开一个房间，之后客户端的连接便提示游戏正在运行，人数达到上限；我也曾想过在一个房间游戏结束后服务器接着进入循环，等待客户端连接，可是目前仍有些未知的错误导致未实现；对于同时开多个房间而不用等待一个房间游戏结束我想可以通过多线程或者多进程来实现。

**⑥总结：**

通过这次石凯老师linux操作系统的学习，我真的感觉我学到了很多知识，无论是对操作系统的理解，还是编程能力的提升，都对我今后的学习有很大的帮助；这次的游戏大作业，也揭开了我对服务器的神秘感，我也对服务器和客户端的通信，多线程多进程的使用和同步，同步异步阻塞非阻塞IO模型的理解都更深了一步；今后我也将努力学习Linux操作系统的知识，认识到它的美。

东北大学

软件学院1507

20154994

周佳男