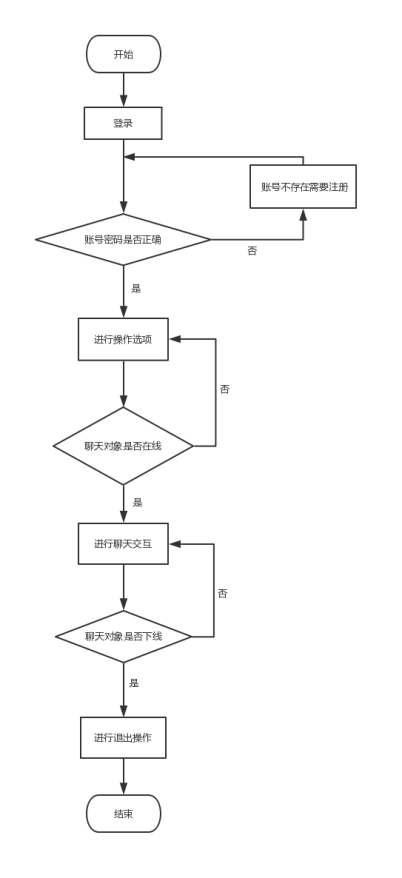
项目标题：go语言完成一个简易聊天软件

参与人员：闫旭

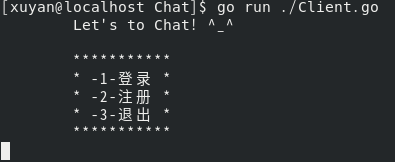
# 项目基本功能描述：

1. >支持账户注册功能（登陆时要用）
2. >支持登录功能（）登录成功才能发消息
3. >客户段与客户端发消息需要服务器转发
4. >支持多个客户端同时在线互相发送消息
5. >客户端收到消息要展示一下内容：谁发的，什么时候发的，发送内容。

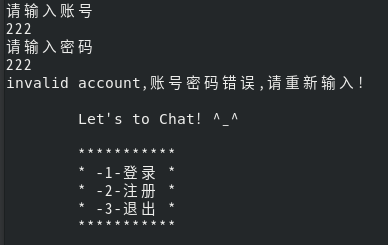
# 功能模块流程介绍：

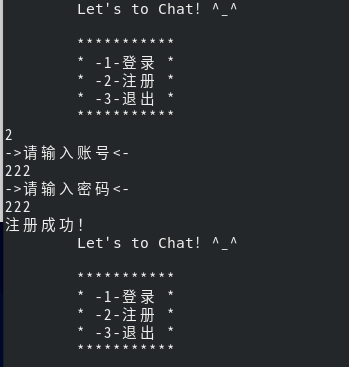


启动程序，展示登陆界面

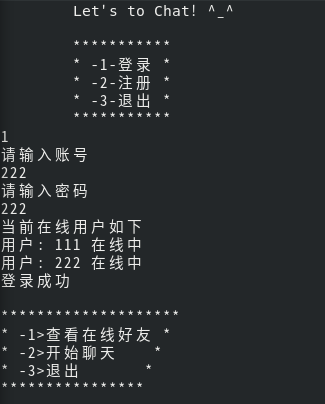


没有账号先注册

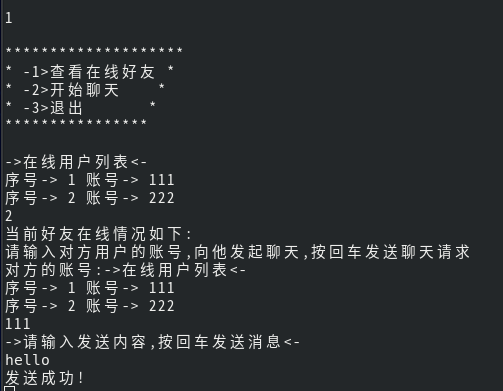




登录程序



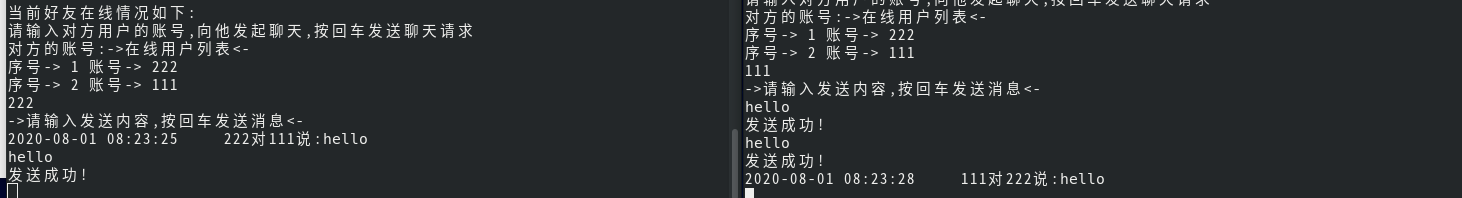
查看在线好友，并向好友发送消息



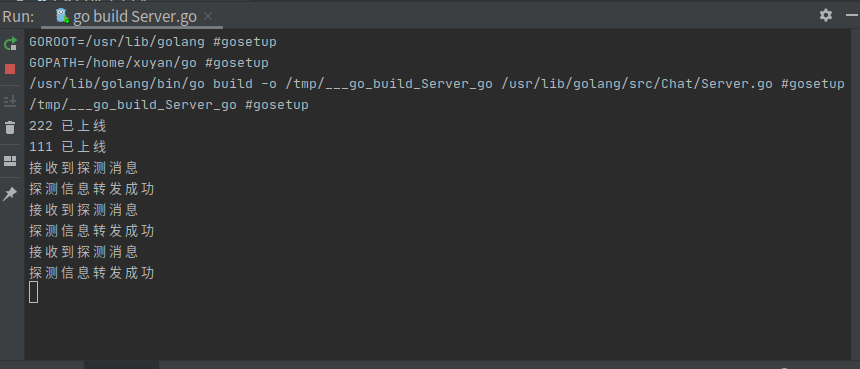
对方在线查看消息



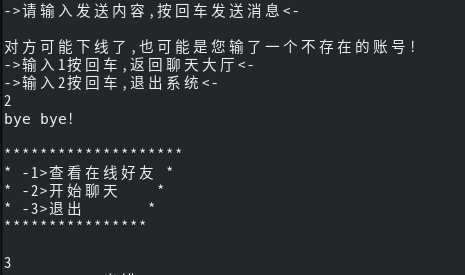
互相发送消息



服务器对消息进行转发



退出程序



# 技术及工具：

go语言、redis高缓存数据库、goland集成开发环境

# 核心代码介绍：

服务端向客户端发送连接请求：

func (this \*userPro) Conn(){

conn,err:=net.Dial("tcp","127.0.0.1:6666")

Common\_tools.HanddleError(err,"连接客户端失败，请检查客户端拨号是否正确")

this.conn=conn

}

客户端监听服务端是否有请求：

func main() {

conn,err := net.Listen("tcp", "0.0.0.0:6666")

Common\_tools.HanddleError(err, "监听错误，请检查端口地址是否正确!")

defer conn.Close()

var transferController=new(Transfer)

var notify=new(Notify)

//挂起协程发送心跳包

notify.HeartBeat()

//通知用户\*\*下线了

notify.NotifyOtherUserOffline()

for {

accept, err := conn.Accept()

Common\_tools.HanddleError(err,"接收数据出错")

go Receive(accept,transferController)

}

}

func (this \*userPro) Login() (error)//登录验证

func (this \*userPro) Register() (error)//注册验证

func Read(accept net.Conn) (read\_str string,read\_slice []byte,n int,err error){

read\_slice=make([]byte,2000)

n,err=accept.Read(read\_slice)

read\_str=string(read\_slice[:n])

return

}//对数据进行读取操作

func Receive(accept net.Conn, transferController \*Transfer)//反序列化函数

# 项目总结：

通过本次项目实训，我学习到了一些go语言得常用语法结构，redis高缓存数据库如何使用，以及对面向对象的编程有了更深入的体会。在项目实操过程中也遇到过大大小小的问题，比如服务端与客户端的接口对接问题，还有工程内的导包问题，redis数据库如何接入工程中等，同过不断地调试修改，最终解决了这些问题，也收获了一些解决小问题的经验。