

# Chat Room Manual

## Part I 系统要求

任意主流Linux 发行版，本 manual 以 Ubuntu（建议16.04及以上版本）为准。

- 首先安装必要工具，进入终端输入

```
sudo apt-get update
sudo apt-get install build-essential vim
```

## Part II 配置Go环境

- 由于一些原因，机房所有的机器的Ubuntu系统都已经具备了Go环境，但是版本较老，你也可以选择直接使用
- Golang中国官方网站：<https://golang.google.cn/>（官方文档以及教程所在）
- Golang软件包下载地址：<https://golang.google.cn/dl/>（目前最新版本1.12.7）
- 下载得到软件包 go1.12.7.linux-amd64.tar.gz
- 进入软件包所在目录打开终端输入

```
tar -C /usr/local -xzf go1.12.7.linux-amd64.tar.gz
sudo vim /etc/profile
```

- 此时进入文本编辑器，键入i进入插入模式，在文件的最后添加

```
export PATH=$PATH:/usr/local/go/bin/
```

- 随后按下esc键，输入 :wq，回车退出
- 随后创建工作目录

```
mkdir $HOME/go
```

- 完成后注销重新登录系统
- 进入用户目录下的go文件夹（也就是之前创建的文件夹），创建文件 `main.go`

```
package main
import "fmt"
func main(){
    fmt.Println("Hello world")
}
```

- 在终端中输入

```
go run main.go
```

如果编译正常并输出Hello World，则表明环境建设成功

## Part III 配置Go集成开发环境 (IDE)

这里介绍Lite IDE 的配置方法，也可以选择你喜欢的开发软件

- 官方网站 <https://github.com/visualfc/liteide/releases/tag/x36>
- 下载 liteidex36.linux64-qt5.5.1.tar.gz
- 进入压缩文件所在目录执行

```
sudo tar -C /usr/local -zxvf liteidex36.linux64-qt5.5.1.tar.gz
```

- 利用与之前相同的方式，在 /etc/profile 文件尾部添加

```
export PATH=$PATH:/usr/local/liteide/bin
```

- 注销并重新登录
- 在终端中输入liteide即可打开IDE

## Part IV 配置Qt5开发环境(可选)

样例代码使用Qt开发环境，Qt是一个流行易用的跨平台C++库，可能会使用到其中有关图形界面，多线程，以及网络相关的库，安装非常简单，打开终端，输入

```
sudo apt-get install cmake qt5-default qtcreator
```

安装将花费比较长的时间

- Qt参考样例 <https://doc.qt.io/qt-5/qtexamplesandtutorials.html>
- Qt文档 <https://doc.qt.io/>

## Part V 服务端

提供样例服务器代码，本代码只有最基本的功能，即将客户端发送来的数据返回给客户端，助教希望你

- 阅读原有代码，并修改或重写代码以实现聊天室服务器基本功能，即能够将用户所发送的消息广播给所有在线用户。
- 修改或重写代码以实现聊天室服务器额外功能，即你想要实现的额外功能。
- 考虑如何利用Go的特性以及系统接口提高服务器性能。

## Part VI 客户端

客户端样例代码使用Qt开发，界面丑陋且原始，助教希望你

- 利用你喜欢的图形界开发工具或语言（GTK, SDL, JS, HTML5 ...）在你喜欢的平台（Windows, Linux, OS X, iOS, Android, Symbian ...）上开发具有基本聊天室功能的客户端
- 或者修改助教的代码，使其具有基本聊天室功能。
- 开发你喜欢的额外功能

## Part VII 评分标准

原则0：组内成员不构成竞争关系。

原则1：鼓励相互帮助

原则2：鼓励自由发挥

### 0.文档

无论你完成结果如何，你都需要提交至少一份文档（5%）

- 开发文档（文本）：假设你要把这个项目转交给下一个开发者，你应当写明什么内容来帮助到他快速展开后续开发？
- 用户说明：你要告诉用户如何使用你的软件，这个说明的形式不限，甚至可以是在程序第一次运行时的提示等等。但你需要证明通过你的用户说明能够保证用户学会使用你的软件。

### 1. 基本分

1.1 能够正确搭建环境，并运行助教代码的（60%）

1.2 在1.1基础上能够解释助教代码工作原理并具有进一步实现的想法的（70%）

1.3-1 在1.2基础上能够实现基本聊天室功能的（80%）

1.3-2 如果你选择不使用助教的代码，则 实现基本聊天室功能（80%）

其中"基本聊天室功能"定义如下

- 具有能够工作的Go语言服务端和客户端
- 具有图形界面（任意平台）
- 具有用户的概念（即用户可以指定使用某个名字进行交流，也可以更换。）（此范围内不要求登陆或注册）
- 用户端能够正常的同时接收发送消息（你需要证明这件事）
- 服务端能够同时服务多个用户（你需要证明这件事）

至此为止，你已经获得了85%的分数

### 2. 奖励分

在完成基本分的基础上你实现的任何额外功能都将获得奖励分，例如

- 建立用户数据库，以达成登陆/注册机制

- 支持发送除文字以外的内容（文件，图片，表情包）等等
- 美化界面
- 优化服务器性能（需要证明）

等等等等，此处不再列举，你可以实现任何你觉得可以提高使用体验的功能。

最终评分将通过你实现功能的复杂程度以及易用性等方面来给定（15%）

### 3.额外分

- 如果你觉得你的作品非常优秀，希望与大家分享，你可以在PPCA的最后通过演讲来展示你的作品，这将获得可以抵充你其他项目上分数的亏空（原则上不多于5分）

最终解释权归PPCA-2019 助教组所有。