****

**计算机网络课程设计**

**报告**

实习任务名称： Project 2A

学 号： 20181003443

姓 名： 孙思思

**中国地质大学地理信息工程学院**

**软件工程系**

**2020年 7 月**

目录

[1 需求简析 2](#_Toc44885054)

[2 Client 2](#_Toc44885055)

[2.1 顶层设计 2](#_Toc44885056)

[2.2 详细设计 3](#_Toc44885057)

[3 Tracker 3](#_Toc44885058)

[3.1 顶层设计 3](#_Toc44885059)

[4 Server 3](#_Toc44885060)

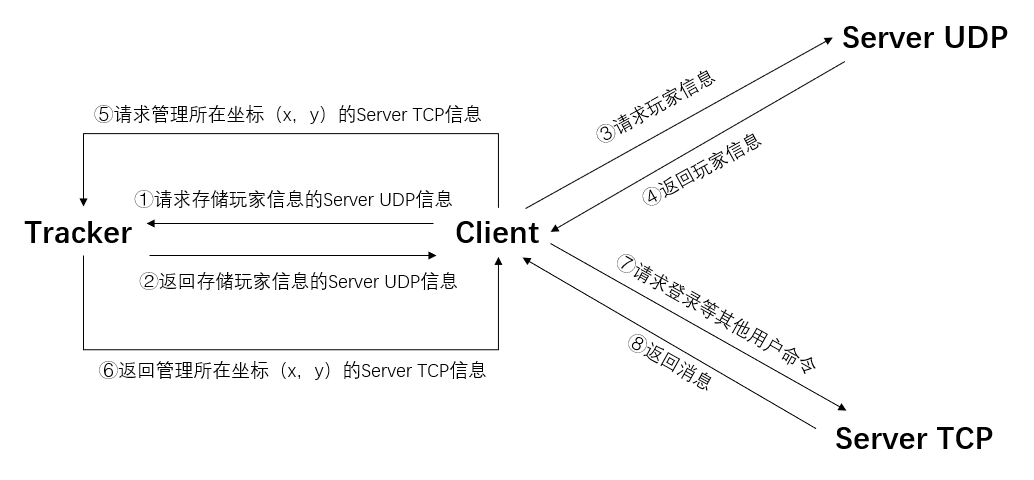
[4.1 顶层设计 3](#_Toc44885061)

[5 测试 4](#_Toc44885062)

[6 总结 4](#_Toc44885063)

# 1 需求简析

当玩家过多时，需要增加服务器来扩展业务。这时客户端用UDP与跟踪器通信，来获取存储玩家信息的服务器IP与UDP端口；然后与此服务器通信，获取玩家数据（HP，EXP，X，Y）；再次与跟踪器通信，获取处理当前位置（X，Y）的服务器IP与TCP端口；至此，可以像project1一样用TCP与服务器通信。（注：部分协议有变动，详情参照2A.Warcraft.doc）



图一 登入流程简图

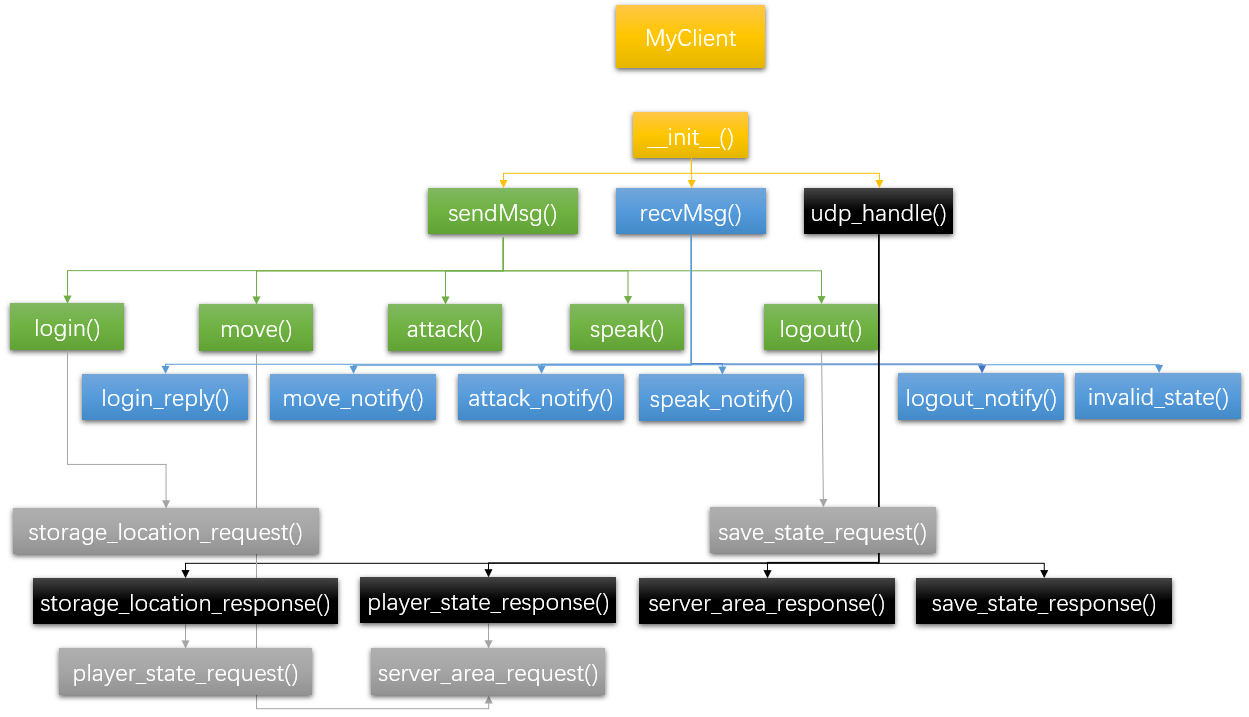


图二 登出流程简图

# 2 Client

## 2.1 顶层设计

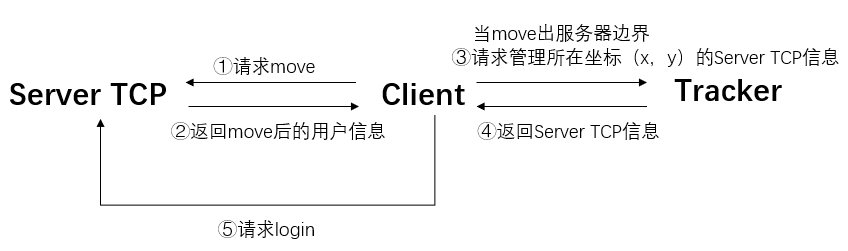
客户端用了两个socket来实现不同需求的通信，一个UDP与Tracker、Server UDP通信，另一个TCP与Server TCP通信，虽然图一把Server的UDP与TCP分开，但是Tracker读取的配置文件中一个Server有UDP端口与TCP端口。



图三 MyClient类的主要函数的依赖关系图

## 2.2 详细设计

在project1的基础上增加了一个新线程（依赖函数udp\_handle），来处理UDP通信的回包。逻辑变动比较大的是login、move、logout，登入登出在需求简析里面分析过了，下面看一下move的不同之处。

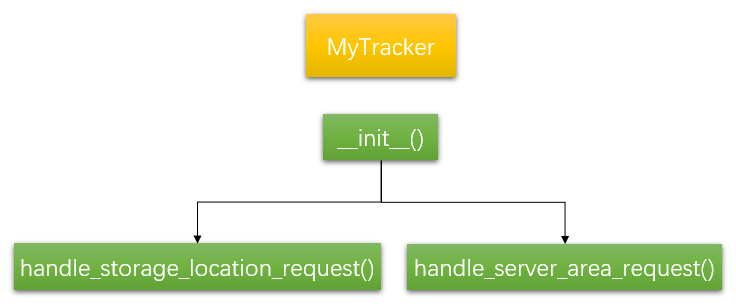


图四 move命令的流程简图

# 3 Tracker

## 3.1 顶层设计

Tracker的功能就两个：一是根据用户请求storage\_location\_request里的名字，用特定的hash函数得到存储其信息的服务器，返回服务器的udp信息给客户端；二是根据用户请求server\_area\_request里面的用户坐标，计算出负责管理这个位置的服务器，返回服务器的TCP信息给客户端。

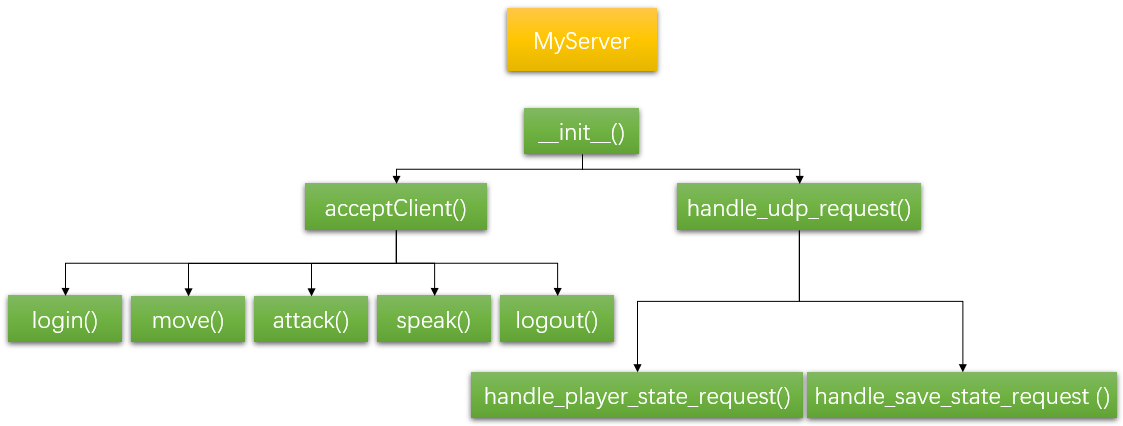


图五 MyTracker的主要函数依赖关系图

# 4 Server

## 4.1 顶层设计

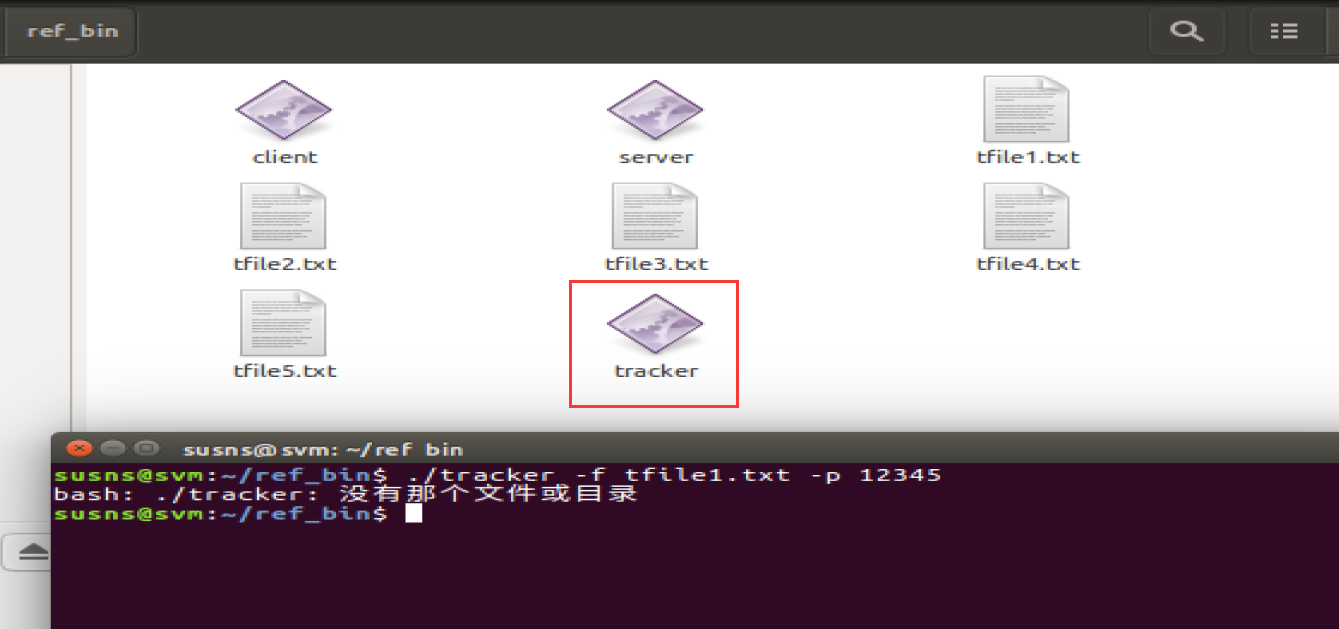
逻辑大部分与project1相似，增加了一个线程（handle\_udp\_request）来处理udp业务，如获取玩家状态、存储玩家状态。



图六 MyServer的主要函数的依赖关系图

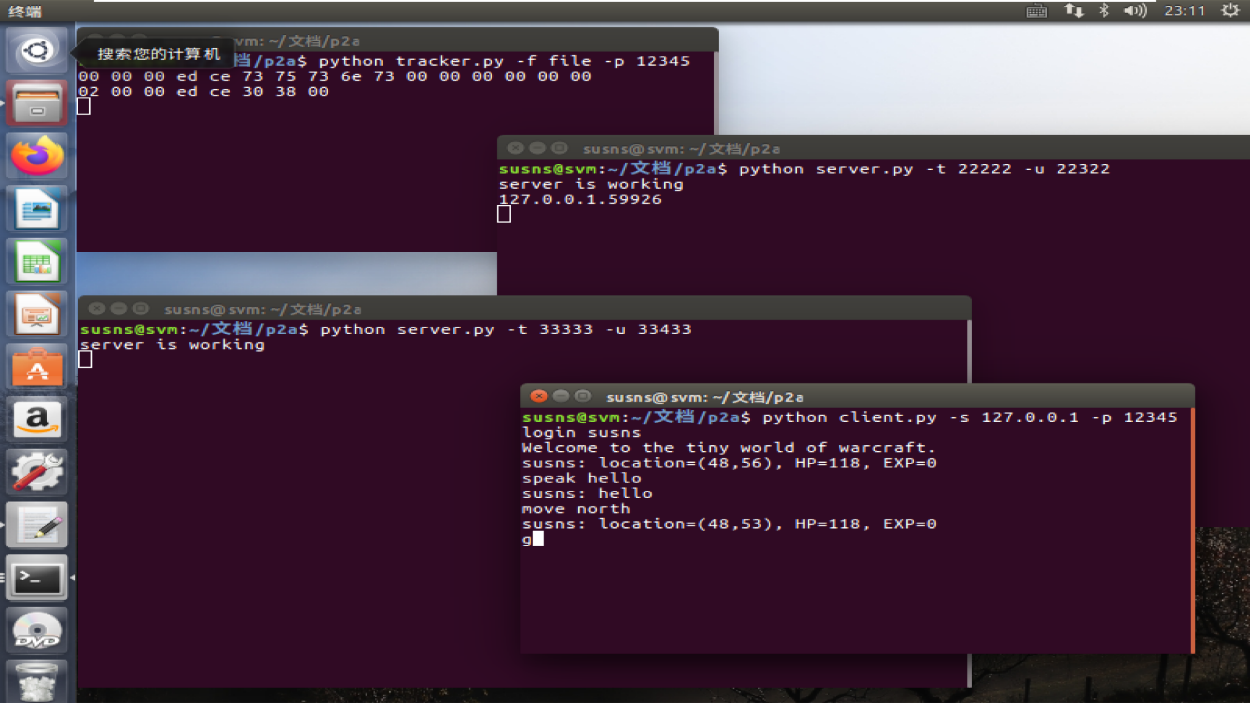
# 5 测试

运行给出的2a的二进制代码失败，所以就没有测试，没有抠输出。



图七 运行给出的2a项目失败图

运行自己的代码，做了简单的测试：



图八 简单测试图

# 6 总结

我觉得本次实习自己最大的进步是对所写代码增加异常的处理，这是以前没有做过的事，自己在慢慢把一份代码写的更加健壮。

存在的不足是：对一份大代码量的实习的规划还不够熟练，准确，以及敲代码的速度还需要提升。