



《计算机网络》

课程设计报告

题目： 20620 游戏软件

班级： 111171

姓名： 王文旭、周宁、王纪峰

指导老师： 罗忠文

中国地质大学（武汉）

地理与信息工程学院软件工程系

2019 年 7 月

一、主要设计思路：

我们选择 Tcp 协议进行传输，之所以采用 tcp, 首先是因为 tcp 传输稳定，相比于 udp 不稳定的传输，tcp 能确保消息一定传输出去。然后是 tcp 有服务器，相比于 p2p 无服务器模型，tcp 可以记录数据，方便用于商用而且也比较安全。

1、客户端主要设计思路：

基于 TCP 协议，实现发送消息给服务器和从服务器接收消息。要实现“迷你魔兽世界”客户端的基本框架主要分为以下几个部分：

(1)、指令，根据输入的指令，比如 login, move, attack 等，客户端要判断指令的有效

指令设计，输入指令 login playername, 客户端向服务器发送登录请求，输入指令 move direction, 客户端向服务器发送移动请求，输入指令 attack playername, 客户端发送攻击另一位玩家的请求，输入指令 speak message, 表示该玩家要发言，内容为 message, 同样要向服务器发送请求，输入指令 logout, 表示退出登陆。

无效指令，如果输入的不是约定好的指令以及指令结构不正确，则客户端不向服务器发送请求。第一，登陆时玩家名不能有空格；第二，move 指令后面的方向必须在 north, south, west 和 east 这几个方向中，如果为其他内容，则输出无效的方向的提示；第三，输入 attack 指令后要判断被攻击玩家是否在该玩家视野范围内，如果不在视野范围内，则输出错误信息“The target is not visible”，玩家视野范围为 (x-5, x+5), (y-5, y+5), x 和 y 为玩家坐标；第四，speak 生成的数据中可以有空格；第五，logout 指令，没有参数。

(2)、发送请求，根据输入的有效指令得到消息类型和消息内容，按照协议打包成字节发送给服务器。

(3)、处理应答，客户端接收到来自服务器的包，根据协议，先解析出消息类型，再根据不同消息类型的包的结构解析出包中的数据，再根据这些数据输出消息。

(4)、线程问题，客户端必须和服务端进行异步通信。例如，当其他玩家生成一个 speak 命令，客户端将会收到 SPEAK_NOTIFY 信息从服务器端，尽管客户端没有请求这个消息。这意味着客户端不能阻塞用户的输入。所以将输入指令发送请求和接收广播/应答的函数设为两个线程，其中一个通过一个死循环，输入指令并处理，另一个也是通过死循环不停接收广播/应答并处理。

(5)、异常处理，当遇到一些特殊情况时需要做异常处理。

如果服务器端不可用并且连接不能生成，客户端必须退出，且显示错误信息“The gate to the tiny world of warcraft is not ready.”。要实现这一点，需要通过 try/except 连接地址和端口号的时候检测异常。

如果客户端从服务器端收到一个异常消息，客户端必须退出，然后输出错误消息“Meteor is striking the world.”。即如果客户端收到的包的长度不符合规范，或者解析到的消息类型不存在或与包的长度不匹配，说明客户端收到了一个异常消息。

如果客户端和服务端端的连接被异常中断，客户端必须退出，然后显示一个错误消息“The gate to the tiny world of warcraft has disappeared.”。由于 python 中的套接字断开连接时不会出现异常，而是会不断接收到空字符，所以在接收消息的地方做一个判断，如果接收到的消息长度为 0，说明连接断开，则执行该异常处理。

2、服务器主要设计思路

服务器主要分为两类 Server 类和 Processor 类，Server 类用来接收客户端的连接，这是主线程，服务器为每一个客户端开辟一个线程 tcplink，每个线程中有一个用来处理线程的对象 processor，用来处理每一个客户端传来的包。主要分为以下几个部分。

接受请求：

服务器接受客户端传来的数据流，然后将数据流解析成一个 packet 对象，分为两种情况来处理。

一：数据流不合法，就是传递过来的数据的格式，并没有按照标准的格式，或者中间的数据是不合法的。这些数据将会被打包成一个 None，然后判断后，断开与服务器的连接，避免是恶意客户端的攻击。

二：数据合法，将 packet 传给 processor，使用 doMessage 函数，将 packet 进行分发，发给同的处理包的函数。如 doLogin（）

请求处理：

处理对象：

处理请求的工作由 processor 来做。Processor 有 curPlayer，由 curPlay 来记录玩家的所有信心，并可以用来判断（curPlay 是否为 None）玩家是否登录，还有两个集合用来记录当前登录的人和注册过的人。另外有 doSpeak，doLogin 等函数，用来处理客户端发来的包。

处理方法：

Login: 当用户登录的时候，先判断客户端的登录状态（当前客户端已登录），玩家名字是否已登录，如果合法，进行登录处理，并向所有已登录玩家发送 LoginReply，MoveNoTify 包，，同时，有已登录的玩家，向当前客户端发送 MoveNotify 包，表明自己的位置。

Move, Speak, Attack: 这三者大同小异，都是处理用户发来的包中的信息，然后进行 Notify。

Loginout: 当玩家退出登录的时候，首先要在服务器的 users 中记录玩家的信息，并保存，然后断开与客户端的连接，并广发 LogoutNotify 包，通知其他玩家，当前玩家已下线。

Notify 处理: 几乎所有的回复都需要进行广播，所以需要使用 broadCast(packet)，这个函数是用来遍历所有的已登录玩家，并发送相应的 Notify 包

HP 按时递增: 为 HP 开辟一个单独线程，按时调用，实现 HP 的按时增加。

异常处理：

异常分为两种：发来的数据错误，客户端的异常中断。

一：数据错误

将与客户端的连接断开。

二：客户端的异常中断

当客户端中断后，客户端会不断发来空字符，所以当收到空字符的时候，就将客户端的连接断开。

3、中间协议包制造和收到的数据处理思路

协议包指按照协议制造的包，协议包方面的设计属于对协议的理解部分，需要正确的理解协议并且按照协议制造这些包。数据处理指客户端或者服务器收到一个消息时，需要对这个消息发送过来的数据进行处理，将其解析成需要的信息，而且需要对消息进行判断，以防有错误的信息。

具体设计：

协议包：

1. 设计两个枚举类，在 Constants.py 文件当中。一个叫 messages，里面的枚举的消息类型，一个叫 direction，枚举移动的方向(可以参照给的 Constants.h)
2. 设计一个基类 Package，里面有 type(包的类型)、ver(版本号)、len(消息的长度)、message(用于套接字发送的字节数据，可能这个不好理解，意思是 sock 发送时，发送的是 sock.send(Package.message))
3. 各种类型协议包继承于 Package，然后里面要添加自己独特的属性，以及 message 的计算。Eg: LOGIN_REQUEST 包，继承于 Package。是客户端登录时发送的请求包，所以按照协议，里面添加了属性 name，并且实现了具体的 message 的制造。

数据处理：

1. 设计 operation 类，里面是各种静态函数用来进行数据处理。
2. operation 里面的 make_packet(), 接收一个 type(包的类型)和*args，进行包的制造，会返回一个调用者想要的 package，如果给定的数据有问题(比如：长度问题、类型不匹配等)，就会返回 None, 方便使用这个函数的人进行判断。
3. operation 里面的 unpack_msg(), 接收一个 msg(是套接字接收的信息，即 msg=sock.recv()), 然后对 msg 进行解析，然后返回相应的 packages(列表，因为 msg 中可能不止一个包)。如：接收的 msg 是关于 login_reply 包，packages 中就包含 login_reply 包，使用时只用 packages[i]. 相应属性即可使用。如果接收的 msg 出现错误(如：长度不对，或者版本号错误)，就会 None, 方便使用者来处理错误信息。

二、遇到的问题解决方法

客户端：

1、问题描述：

程序中设计的函数：run_send(), 通过一个死循环，不断地接收用户输入，向服务器发送包，用户未输入时阻塞；函数 run_receiv(), 通过一个死循环不断接收服务器发来的包，然后解析包并处理，未接收到包时会阻塞。在创建多线程的时候总会有一个线程创建失败。

问题解决方法：

查阅资料，发现在 python 中将函数作为参数创建多线程时，如果参数中的函数带括号的话，创建线程的时候就会先执行一次，然后就会造成阻塞，主线程被阻塞将导致后面的线程创建失败，所以将函数作为参数创建多线程的时候去掉函数名后面的括号之后，问题就得到了解决。

2、问题描述:

当结束进程时，线程没有结束，线程依然在继续运行。原因是进程结束会等待线程退出，但是由于线程是设计成死循环的，所以线程无法正常退出，就会出现错误。

问题解决方法:

创建线程的时候将线程设置为守护线程，如果一个线程是守护线程，则进程退出时不必等待线程退出。

3、问题描述:

当客户端和服务器的连接异常中断，比如服务器突然崩溃，客户端没有收到任何异常消息，无法判断连接状态。

问题解决方法:

查阅资料之后明白在 python 中如果套接字的连接断开的话，客户端执行接收服务器端发来的包的时候会接收到空字符，所以在接收包的地方添加一个判断语句，判断接收的数据长度是否为 0，如果为 0，则说明连接已断开，则结束进程。

4、问题描述:

当用户登陆的时候，服务器端发送了两个或两个以上的包给这个客户端，但是客户端仅仅处理了一个包，剩下的没有处理。

问题解决方法:

经过调试发现登陆时服务器端发来的几个包是一起发的，所以在客户端只需要接收一次就可以接收全部发来的包，而解析时没有注意包的长度，只解析了第一个包，所以在解析包时候要将所有的包也一起解析了，同时，经过测试，我们发现了在一个玩家杀死另一个玩家时也会出现这种情况，服务器强将 ATTACK_NOTIFY 消息和 MOVE_NOTIFY 消息一起发给了客户端。

5、问题描述:

在通过测试 test1a.py 的时候发现我客户端已经输出了内容，但是测试时要获取我输出内容时出现异常。

问题解决方法: 查阅资料之后了解到，在 linux 上后台运行 python 程序 stdout 会先输出到缓冲区，等缓冲区满或者脚本结束再输出，而 stderr 不会先输出到缓冲区。在每一次输出后面添加一行代码清除缓冲区，这样就可以正常接收客户端输出的内容了。

服务器端:

1. 问题描述:

当客户端同时接受到多个 client 发来的请求的时候，服务器无法同时接受这些消息，根据查找，是因为在客户端当中 receddata() 时，由于有其他的客户在输入，这个会阻塞住。所以问题就是处理 recevedata 的阻塞。

问题解决方法: 当服务器端在接受客户的输入的时候，做一个异常检测 (try)，如果抛出异常，就继续循环。

2. 问题描述:

当来自服务器端的客户突然断开连接的时候，服务器会因为连接断开，不断受到空的字节，导致服务器崩溃

问题解决方法: 当收到来自客户端的空字符的时候，判断这个客户端已经断开连接，这个时候，必须断开与服务器的连接，同时要跳出这个线程，删除这个线程的 processor。防止其他客户端向当前客户端发消息。

3. 问题描述:

发送 Notify 包的时候, 需要向所有的客户端广播, 这个时候需要向其他的线程通信。如何处理线程之间的通信。

问题解决方法: 将每一个客户端的处理在主线程中创建出来, 并记录在 pros(集合)当中, 这样客户端可以通过处理者来进行相互之间的通信

4. 问题描述:

2MSL 问题, 当 TCP 主动断开连接之后, 由于需要完成四次握手, 所以进入了 TimeWait 等待状态, 先前的端口就不能使用。

问题解决方法:

使用 setsockopt 设置与套接字关联的选项, 设置为 SO_REUSEADDR (即允许重用本地地址与端口), 这样即使在关闭一个服务器之后, 不需要等待, 就可以直接进行连接了。

协议包制造和数据处理:

1、问题描述:

对于 32 位对齐并没有弄的明白

问题解决方法:

看标准客户机和服务器连接打印的包逐渐明白了中间的数据的组织过程。采用字节来处理, 一个字节 8 位, 32 位对齐即 4 字节对齐即可。

2、问题描述:

对于协议理解没有理解透彻, 然后进行 test1b 测试时发现, speak 包编写时发生了错误, 测试 254 字节时, 我发送过去的长度是 268 字节, 而标准的发送过去是 272 字节。

问题解决方法:

原来是 speak 包在 message 制造时, 我开始以为前面包头 4 字节, 名字是 10 字节, 然后如果 speak 254 字节的内容时。三者相加是 268 刚好是 32 位对齐的, 但是标准的是名字 10 字节补 2 个字节对齐, 然后 254 字节的内容再补 2 字节对齐, 且都补在末尾, 所以多 4 字节。(虽然个人感觉我这样写更好, 符合协议且节约字节, 但是还是更改了过来)

3、问题描述:

解析 msg 时开始以为 msg 只有 1 个包的内容, 结果 msg 包含很多包的内容

问题解决方法:

采用 while 循环逐个解析, 然后将解析出来的包以列表的形式返回

4、问题描述:

在写 speak 包时没注意是可打印这个细节, 以及玩家名字只能是字母和数字这个细节, 测试时发生错误

问题解决方法:

在解析包和制造包时加上判断的函数即可, python 有自带的函数, 可以检测字符串是否是可打印以及字符串是否只包含字母和数字。

三、成果展示

客户端:

Login 和 move 功能展示:

```
command> Player mario has left the tiny world of warcraft.
command>
command> peach damaged mario by 19. mario's HP is now 201.
command> attack mario
command> The target is not visible.
command> peach: this is peach.
command> speak this mario
command> mario: this mario.
command> logout
command> The gate to the tiny world of warcraft has disappeared.
command>
Process finished with exit code 0
```

Speak 功能展示:

```
command> speak this is peach
command> peach: this is peach
command> mario: this mario
command>
command> move west
command> mario: location=(20,17), HP=215, EXP=17.
command> attack peach
command> mario damaged peach by 12. peach's HP is now 170.
command> peach damaged mario by 19. mario's HP is now 201.
command> attack mario
command> The target is not visible.
command> peach: this is peach.
command> speak this mario
command> mario: this mario.
command>
```

Attack 功能展示:

```
command> mario damaged peach by 12. peach's HP is now 170
command> attack mario
command> peach damaged mario by 19. mario's HP is now 201
command>
command> peach: this is peach.
command> move east
command> mario: location=(23,17), HP=214, EXP=17.
command> move west
command> mario: location=(20,17), HP=215, EXP=17.
command> attack peach
command> mario damaged peach by 12. peach's HP is now 170.
command> peach damaged mario by 19. mario's HP is now 201.
command> attack mario
command> The target is not visible.
command>
```

Invalid 功能展示:

```
/home/wwx/PycharmProjects/WoW/venv/bin/python /home/v
command> move west
command> You must log in first.
command>
```

Logout 功能展示:

```
command> Player mario has left the tiny world of warcraft.
command>
command> peach damaged mario by 19. mario's HP is now 201.
command> attack mario
command> The target is not visible.
command> peach: this is peach.
command> speak this mario
command> mario: this mario.
command> logout
command> The gate to the tiny world of warcraft has disappeared.
command>
Process finished with exit code 0
```

跨机测试（跟自己完成的服务器）:

```
wwx@wwx-ZHAOYANG-E42-80: ~/PycharmProjects/WoW/Test
wwx@wwx-ZHAOYANG-E42-80:~/PycharmProjects/WoW/Test$ ./client -s 192.168.43.22 -p
8080
command> login wwx
command> Welcome to the tiny world of warcraft.
command> wwx: location=(75,41), HP=113, EXP=0.
command> move west
command> wwx: location=(72,41), HP=114, EXP=0.
command>
```

Test1a 测试:

```
Run: test1a(1) x
your client passes all 20 command test cases, and 35 of 35 server reply tests.
Now testing exception handling.
Testing connect failure: PASS
Testing server disconnect: PASS
Testing malformed packet case: an invalid version: PASS
Testing malformed packet case: a truncated last packet: PASS
Testing malformed packet case: a packet not 4byte aligned: PASS
Testing malformed packet case: an invalid player name: PASS
Testing malformed packet case: an invalid speak message: PASS
Testing malformed packet case: an invalid location: PASS
npass: 20
rpass: 35
fpass: 8
Points: 100.0

Process finished with exit code 0
```


服务器:

先展示两个客户端的指令:

```
zhouning@zhouning-RESCUER-R720-15IKBN: ~/桌面/TWoW
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
zhouning@zhouning-RESCUER-R720-15IKBN: ~/桌面/TWoW$ ./client -p 8080 -s 127.0.0.1
command> login zn
command> Welcome to the tiny world of warcraft.
command> zn: location=(82,43), HP=192, EXP=0
command> login zn
command> You already logged in.
command> move north
command> zn: location=(82,40), HP=278, EXP=0
command> wjf: 1111
command> █

zhouning@zhouning-RESCUER-R720-15IKBN: ~/桌面/TWoW
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
zhouning@zhouning-RESCUER-R720-15IKBN: ~/桌面/TWoW$ ./client -p 8080 -s 127.0.0.1
command> login wjf
command> Welcome to the tiny world of warcraft.
command> wjf: location=(37,62), HP=146, EXP=0
command> move north
command> wjf: location=(37,59), HP=226, EXP=0
command> speak 1111
command> wjf: 1111
command> █
```

登陆展示:

```
Run: server
/home/zhouning/anaconda3/bin/python3.7 /home/zhouning/PycharmProjects/wow/Server/server.py
port:8080
* Listening socket is ready.
New connection from 127.0.0.1.36900. fd=4
-fd:4 received msg ver:4 len:16 type:1 raw_pkt(net_byte_order)=[04 00 10 01 7a 6e 00 00 00 00 00 00 00 00 00 00 ]
-fd:4 sending msg ver:4 len:16 type:2 raw_pkt(net_byte_order)=[04 00 10 02 00 00 00 00 c0 00 00 00 00 52 2b 00 ]
-fd:4 sending msg ver:4 len:24 type:4 raw_pkt(net_byte_order)=[04 00 18 04 7a 6e 00 00 00 00 00 00 00 00 52 2b 00 00 00 c0 00 00 00 00 ]
New connection from 127.0.0.1.36902. fd=5
-fd:5 received msg ver:4 len:16 type:1 raw_pkt(net_byte_order)=[04 00 10 01 77 6a 66 00 00 00 00 00 00 00 00 00 ]
-fd:5 sending msg ver:4 len:16 type:2 raw_pkt(net_byte_order)=[04 00 10 02 00 00 00 00 92 00 00 00 00 25 3e 00 ]
-fd:5 sending msg ver:4 len:24 type:4 raw_pkt(net_byte_order)=[04 00 18 04 77 6a 66 00 00 00 00 00 00 00 25 3e 00 00 00 92 00 00 00 ]
-fd:4 sending msg ver:4 len:24 type:4 raw_pkt(net_byte_order)=[04 00 18 04 77 6a 66 00 00 00 00 00 00 00 25 3e 00 00 00 92 00 00 00 ]
-fd:5 sending msg ver:4 len:24 type:4 raw_pkt(net_byte_order)=[04 00 18 04 7a 6e 00 00 00 00 00 00 00 00 52 2b 00 00 00 00 00 00 00 ]
```

重复登陆展示:

```
-fd:4 sending msg ver:4 len:16 type:1 raw_pkt(net_byte_order)=[04 00 10 01 7a 6e 00 00 00 00 00 00 00 00 00 00 ]
-fd:4 received msg ver:4 len:16 type:1 raw_pkt(net_byte_order)=[04 00 10 01 7a 6e 00 00 00 00 00 00 00 00 00 00 ]
-fd:4 sending msg ver:4 len:8 type:11 raw_pkt(net_byte_order)=[04 00 08 0b 01 00 00 00 ]
|
```

向北移动展示 (其中的 0 代表方向):

```
Run: server
-fd:5 received msg ver:4 len:8 type:3 raw_pkt(net_byte_order)=[04 00 08 03 00 00 00 00 ]
0
-fd:4 sending msg ver:4 len:24 type:4 raw_pkt(net_byte_order)=[04 00 18 04 77 6a 66 00 00 00 00 00 00 00 25 3b 00 00 00 e2 00 00 00 00 ]
-fd:5 sending msg ver:4 len:24 type:4 raw_pkt(net_byte_order)=[04 00 18 04 77 6a 66 00 00 00 00 00 00 00 25 3b 00 00 00 e2 00 00 00 00 ]
|
```

客户端 attack:

```
zhouning@zhouning-RESCUER-R720-15IKBN: ~/桌面/TWoW
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
zhouning@zhouning-RESCUER-R720-15IKBN:~/桌面/TWoW$ ./client -p 8080 -s 127.0.0.1
command> login wjf
command> Welcome to the tiny world of warcraft.
command> wjf: location=(10,10), HP=277, EXP=0
command> zn: location=(12,12), HP=352, EXP=0
command> attack zn
command> wjf damaged zn by 11. zn's HP is now 346
command> 
command> move north
command> zn: location=(82,40), HP=278, EXP=0
command> wjf: 1111
command> Player wjf has left the tiny world of warcraft.
command> logout
command> The gate to the tiny world of warcraft has disappeared.
zhouning@zhouning-RESCUER-R720-15IKBN:~/桌面/TWoW$ ./client -p 8080 -s 127.0.0.1
command> login zn
command> Welcome to the tiny world of warcraft.
command> zn: location=(12,12), HP=347, EXP=0
command> wjf: location=(10,10), HP=277, EXP=0
command> wjf damaged zn by 11. zn's HP is now 346
command> |
```

服务器端:

```
Run: server
- fd:5 received msg ver:4 len:16 type:5 raw_pkt(net_byte_order)=[04 00 10 05 7a 6e 00 00 00 00 00 00 00 00 00 00 ]
- fd:4 sending msg ver:4 len:32 type:6 raw_pkt(net_byte_order)=[04 00 20 06 77 6a 66 00 00 00 00 00 00 7a 6e 00 00 00 00 00 00 0b 00 00 01 5a 00 00 00 ]
- fd:5 sending msg ver:4 len:32 type:6 raw_pkt(net_byte_order)=[04 00 20 06 77 6a 66 00 00 00 00 00 00 7a 6e 00 00 00 00 00 00 0b 00 00 01 5a 00 00 00 ]
```

Speak 展示:

```
Run: server
- fd:5 received msg ver:4 len:8 type:7 raw_pkt(net_byte_order)=[04 00 08 07 31 31 31 31 ]
- fd:4 sending msg ver:4 len:20 type:8 raw_pkt(net_byte_order)=[04 00 14 08 77 6a 66 00 00 00 00 00 00 31 31 31 31 00 00 ]
- fd:5 sending msg ver:4 len:20 type:8 raw_pkt(net_byte_order)=[04 00 14 08 77 6a 66 00 00 00 00 00 00 31 31 31 31 00 00 ]
```

Logout 展示 (只有一台客户端时):

```
Run: server
- fd:4 received msg ver:4 len:4 type:9 raw_pkt(net_byte_order)=[04 00 04 09 ]
- Garbage collected. fd=4
|
```

跨机连接 (和自己写的客户端):

```
Run: server
/home/zhouning/anaconda3/bin/python3.7 /home/zhouning/PycharmProjects/wow/Server/server.py
port:8080
* Listening socket is ready.
New connection from 192.168.43.103.36492. fd=4
- fd:4 received msg ver:4 len:16 type:1 raw_pkt(net_byte_order)=[04 00 10 01 77 77 78 00 00 00 00 00 00 00 00 00 ]
- fd:4 sending msg ver:4 len:16 type:2 raw_pkt(net_byte_order)=[04 00 10 02 00 00 00 00 6e 00 00 00 00 22 18 00 ]
- fd:4 sending msg ver:4 len:24 type:4 raw_pkt(net_byte_order)=[04 00 18 04 77 77 78 00 00 00 00 00 00 22 18 00 00 6e 00 00 00 00 ]
```

Test1b 测试:

```
***** Slow test (login+order+logout): PASS
* Duplicated login test: PASS
* Move command test: PASS
* HP regeneration test: PASS
* Speak command test (100 bytes + null): PASS
* Speak command test (254 bytes + null): PASS
* Now testing the attack command.
* This may take long since we move players.
Player1 loc: (65,42)
Player2 loc: (91,57)
Player1 moves east 8 times
Player1 moves south 5 times
* Attack command test.: PASS
* Logout command test: PASS
* Player data persistency test (normal logout): PASS
* Player data persistency test (client crash): PASS
* Exception test (invalid version): PASS
* Exception test (invalid length): PASS
* Exception test (invalid msg type): PASS
* Exception test (invalid move direction): PASS
* Exception test (invalid player name): PASS
* Exception test (invalid speak message): PASS
! Exception test (emulating a slow link): FAIL w/ an exception: # SPEAK NOTIFY is not 1. Got 0. Client output:  enqueued msg ver:4 len:16 type:1 raw_pkt(net_byte_o
received msg ver:4 len:16 type:2 raw_pkt(net_byte_order)=[04 00 10 02 00 00 00 00 70 00 00 00 00 41 24 00 ]
Welcome to the tiny world of warcraft.
received msg ver:4 len:24 type:4 raw_pkt(net_byte_order)=[04 00 18 04 6d 61 72 69 6f 00 00 00 00 00 41 24 00 00 00 70 00 00 00 00 ]
mario: location=(65,36), HP=112, EXP=0
* Fault "slow link" is on.
enqueued msg ver:4 len:56 type:7 raw_pkt(net_byte_order)=[04 00 38 07 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36 37 38 39 30 31 :
The gate to the tiny world of warcraft has disappeared.

* Exception test (truncated last packet): PASS
* Performance test (speak messages by 5 clients): PASS
* Performance test (speak messages by 10 clients): PASS
* Performance test (speak messages by 20 clients): PASS
* Finished all the tests.
Function tests (13): 130/130
Exception tests (8): 70/80
Performance tests (3): 30/30
Earned 230 out of 240
Points: 95.8333333333
```

注：Test1B 的测试目前仍然有两个问题：第一是无法处理慢速连接攻击。第二是测试的时候必须使用 Debug 模式，就是进行逐步调试，才能出现 PASS，否则的话，客户端启动不成功。

四、实习收获和心得体会

王文旭（20171003247）：

主要任务：负责客户端的编写

收获：

客户端要做的事情并不是特别多，遇到的问题主要是进程和读写缓冲区方面的一些知识，在实习过程中也学到了许多 python 以及 TCP 连接的知识。在实习中将客户端的处理数据的部分独立出来，由另一位队友负责实现，约定好一些接口，客户端调用这些接口，不用管这些函数是如何实现，由此来提高效率，所以实习过程中团队合作的经验对我来说也是很大的收获。

周宁（20171004140）：

主要任务：编写协议包和数据处理

收获：

先说说实习外的收获，这是我第一次在 Linux 环境下进行编程，开始安装双系统时还浪费了一些时间，不过体验非常棒，作为一个软工的学生学会使用双系统应该是必备技能。然后这也是我第一跟别人合作一起写代码，以前我都是一个人独立战斗，这次是三个人一起，虽然中途的时候也有不好的体验，但是总体说来还是不错的，为以后和别人一起开发提供了经验。然后对于这次实习，我也有许多的收获，首先是编写代码测试的时候一定要找对方法，这样可以达到事半功倍的效果。然后对于计算机网络中的协议，我也有了自己的看法，其实协议也并不是什么“高大上”的东西，其实有点类似于约定一样，只要你设计的足够细致，你也能够创造出协议。

王纪锋 (20171002130):

主要任务: 服务器的编写

收获:

在这次的编写过程中,使我收获最多的还是对于线程的使用更加的了解,线程之间通信,已经所有的线程之间的数据的同步,让我收获巨大。还有 TCP 编程中的一些问题,我也开始了解了, TCP 的使用过程中虽然是可靠地,但还是有很多的问题需要自己逐步解决。以及对于各种各样的异常处理,也让我对以后代码编写过程中,测试的范围更加的了解,不仅要有各种正确的指令的处理,还要有各种异常的,非法的数据也要进行处理。

可以改进的地方:

当前的标准服务器与客户端,处理玩家的信息全部交给了客户机来处理(如:当一个玩家移动的时候,其他所有的玩家都会收到这个玩家的消息,这样它们就知道了这个玩家的信息),这样一个客户端就可以获取到其他的所有的玩家的信息。这样处理是一种不安全的行为。玩家信息处理应该放在服务器端进行,有选择的发送消息,例如只给视线范围内的玩家发送消息。首先,当前情况下,服务器已经存储了所有玩家的消息,我们可以对玩家的信息增加一些判断(如其余玩家是否在当前玩家视野范围内,如果在的话,在进行消息的发送)。