

多人聊天室及文件共享平台

- 组长: 施俊杰
- 组员: 李焱阳 蔡瑛珈 林日荣

人员分工

施俊杰负责程序的实现以及报告文档的撰写，李焱阳负责展示和PPT制作，林日荣和蔡瑛珈讨论了很多技术上的细节并设想了一些功能。

程序概况

- 本程序用Python2.7完成，程序共分为客户端（clin.py）和服务端（serv.py）两部分。两边都主要使用多线程的编程方法。
- 主要的思想，就是服务器端和客户端创立Socket，然后建立TCP连接，之后便可以传输数据。只不过服务器端必须使用多线程的方式来同时监听，还有处理多个并发的连接。客户端也必须用两个线程来处理可能会并发的发送数据与接收数据的情况。clin.py和serv.py中都有程序的详细解释，本文档是概括性的解释思路以及程序的用法。
- 功能有二：
 - **多人聊天室**：类似QQ群，多个客户端连接服务器后，每台机器所发的消息其他人都能接收，实现多人聊天。
 - **文件共享**：依然类似QQ群的文件功能，也类似Git，每个客户端可以把文件发送到服务器上，同时也可以请求下载服务器端的文件，实现文件共享功能。
- 特殊命令：同时程序具有一些特殊的命令行功能，当输入这些特殊命令时将产生特殊操作，服务器只对发出命令的客户端响应，对其他用户来说察觉不到。

所有的命令都以'!'开头，主要是查询和服务器的上传下载功能，具体有：

!bye:	客户端断开连接
!howmany:	查询当前有多少用户，服务器将返回当前人数
!users:	查询当前的用户列表，即所有人的昵称
!ls:	查询当前服务器端的文件目录
!get [file name]:	从服务器下载某文件到本地
!ft:	传输文件（file transportation）指令，输入指令后将开始指定文件名，输入文件名成功后可将文件上传到服务器

多人聊天室

程序比较直观。首先需要运行 `serv.py` 来启动服务器，输入服务器所用的IP地址，之后服务器便进入监听状态：

```
C:\Users\Administrator\Desktop\网络大作业报告>python27 serv.py
input the server's ip address: localhost
Listening...
```

每个客户端在连接服务器时，为自己指定一个name作为昵称（同时这个昵称不能与当前服务器中已存在昵称重复，否则将要求重新输入昵称，直到没有重复昵称为止）。然后输入希望连接的服务器的IP地址。如果以上两项无误，就成功进入聊天室。

```
C:\Users\Administrator\Desktop\网络大作业报告>python27 clin.py
input your name: No.1
input the server's ip address: localhost
the name No.1 has been used, use another one!
No.2
Welcome No.2 to the room!
```

多个客户端之间可以实现简单的聊天功能。

```
Microsoft Windows [版本 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd neo

C:\Users\Administrator\neo>python27 clin.py
input your name: No.1
input the server's ip address: localhost
Welcome No.1 to the room!
Welcome No.2 to the room!
No.2: Hi I'm No.2
I am No.1
No.1: I am No.1
```

```
C:\windows\system32\cmd.exe - python27 clin.py

C:\Users\Administrator\neo>python27 clin.py
input your name: No.2
input the server's ip address: localhost
Welcome No.2 to the room!
Hi I'm No.2
No.2: Hi I'm No.2
No.1: I am No.1
```

!howmany，当前人数的查询，只对本机响应：

```
C:\windows\system32\cmd.exe - python27 clin.py

Microsoft Windows [版本 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd neo

C:\Users\Administrator\neo>python27 clin.py
input your name: No.1
input the server's ip address: localhost
Welcome No.1 to the room!
Welcome No.2 to the room!
No.2: Hi I'm No.2
I am No.1
No.1: I am No.1
```

```
C:\windows\system32\cmd.exe - python27 clin.py

C:\Users\Administrator\neo>python27 clin.py
input your name: No.2
input the server's ip address: localhost
Welcome No.2 to the room!
Hi I'm No.2
No.2: Hi I'm No.2
No.1: I am No.1
!howmany
2 person(s)!
```

!user，当前用户列表，只对本机响应：

```
Microsoft Windows [版本 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd neo

C:\Users\Administrator\neo>python27 clin.py
input your name: No.1
input the server's ip address: localhost
Welcome No.1 to the room!
Welcome No.2 to the room!
No.2: Hi I'm No.2
I am No.1
No.1: I am No.1
!user
['No.2', 'No.1']
```

```
C:\windows\system32\cmd.exe - python27 clin.py

C:\Users\Administrator\neo>python27 clin.py
input your name: No.2
input the server's ip address: localhost
Welcome No.2 to the room!
Hi I'm No.2
No.2: Hi I'm No.2
No.1: I am No.1
!howmany
2 person(s)!
```

!bye，断开连接，其他用户可看到离线消息：No.2先断开连接，然后No.1断开连接：

```
C:\windows\system32\cmd.exe
Microsoft Windows [版本 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd neo

C:\Users\Administrator\neo>python27 clin.py
input your name: No.1
input the server's ip address: localhost
Welcome No.1 to the room!
Welcome No.2 to the room!
No.2: Hi I'm No.2
I am No.1
No.1: I am No.1
!users
['No.2', 'No.1']
No.2 leaves the room! There are/is 1 people left!
!bye

C:\Users\Administrator\neo>

C:\Users\Administrator\neo>python27 clin.py
input your name: No.2
input the server's ip address: localhost
Welcome No.2 to the room!
Hi I'm No.2
No.2: Hi I'm No.2
No.1: I am No.1
!howmany
2 person(s)!
!bye

C:\Users\Administrator\neo>
```

服务器端可看到:

```
C:\Users\Administrator\neo>python27 serv.py
input the server's ip address: localhost
Listening...
Connected with 127.0.0.1:50905
Welcome No.1 to the room!
1 person(s)!
Connected with 127.0.0.1:50907
Welcome No.2 to the room!
2 person(s)!
No.2: Hi I'm No.2
No.1: I am No.1
No.2 leaves the room! There are/is 1 people left!
No.1 leaves the room! There are/is 0 people left!
```

实现的办法就是一个线程负责接收，一个线程负责发送，二者并行发生，在客户端和服务端都是如此，具体的解释可参看程序，有详细解释。

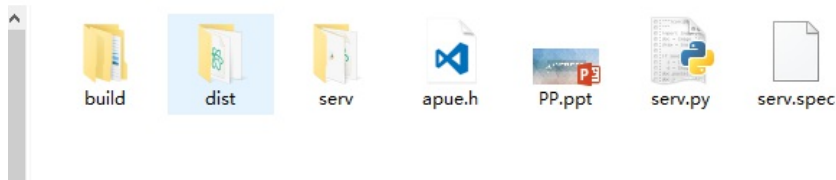
文件共享

主要由上传和下载两个功能组成，同时辅助性的还有一个 `!ls` 来查看当前服务器的文件列表，以知道有哪些可下载的文件。

`!ls`:

这是服务器端的文件：

此电脑 > Win 10 Ent x64 (C:) > 用户 > Administrator > neo

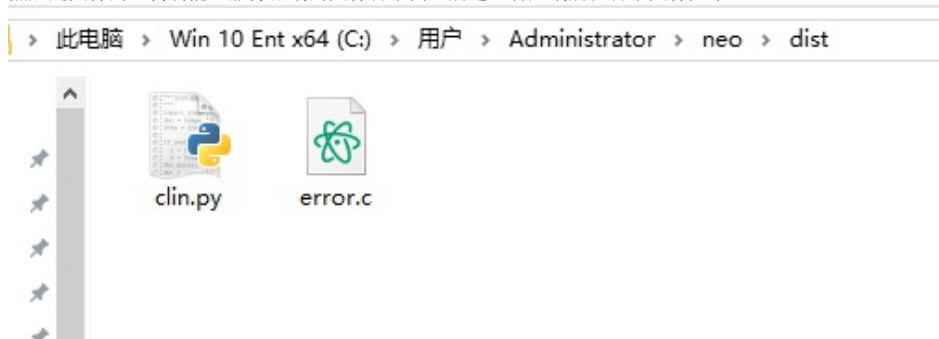


查询结果如下：

```
C:\Users\Administrator\neo\dist>python27 clin.py
input your name: Jack
input the server's ip address: localhost
Welcome Jack to the room!
!ls
apue.h  PP.ppt  serv.py serv.spec
```

只显示文件单位，不显示文件夹。实际上就是在服务器端对关键字做一个检测，当检测到关键字时，调用python的 `os.listdir(os.getcwd())` 功能，并用 `if os.path.isfile(each_file)` 判断，形成一个符合条件的字符串，返回给请求的客户端。当然这也是只对单个客户端生效的，其他客户端看不到。

然后是文件的上传功能，服务器端的文件目录同上所述，客户端所在目录文件如下：



将客户端本地的error.c文件上传到服务器端：

```
C:\Users\Administrator\neo\dist>python27 clin.py
input your name: Jack
input the server's ip address: localhost
Welcome Jack to the room!
!ls
apue.h PP.ppt serv.py serv.spec
!ft
Entre the file name:
SERVOK
error.c
Calculating MD5...
Calculating success
Sending data...
Send success!
MD5 : 376a1b750a882448d58b533faad04a98
Cost 0.000000 seconds
!ls
apue.h error.c PP.ppt serv.py serv.spec
```

输入 `!ft` 后可看到服务器端发来的SERVOK信号，此时便可输入要上传的文件名。同时上传完成后在服务器端也能看到结果，还有两个用于比较的MD5码：

```

C:\Users\Administrator\neo>python27 serv.py
input the server's ip address: localhost
Listening...
Connected with 127.0.0.1:50467
Welcome Jack to the room!
1 person(s)!
Connected with 127.0.0.1:50472
Welcome Rose to the room!
2 person(s)!
Rose leaves the room! There are/is 1 people left!
Jack leaves the room! There are/is 0 people left!
Connected with 127.0.0.1:50558
Welcome Jack to the room!
1 person(s)!
Waiting file from Jack...
Receiving data...
Accept success!
Calculating MD5...
  Recevie MD5 : 376a1b750a882448d58b533faad04a98
Calculate MD5 : 376a1b750a882448d58b533faad04a98

```

然后是下载功能，比如我们希望下载客户端没有的PPppt文件（客户端和服务器端的文件都如上所述）。那么就用 `!get PP.ppt` 指令：

```

!ls
apue.h error.c PP.ppt serv.py serv.spec
!get PP.ppt
SERVOK
Receiving data...
Accept success!

```

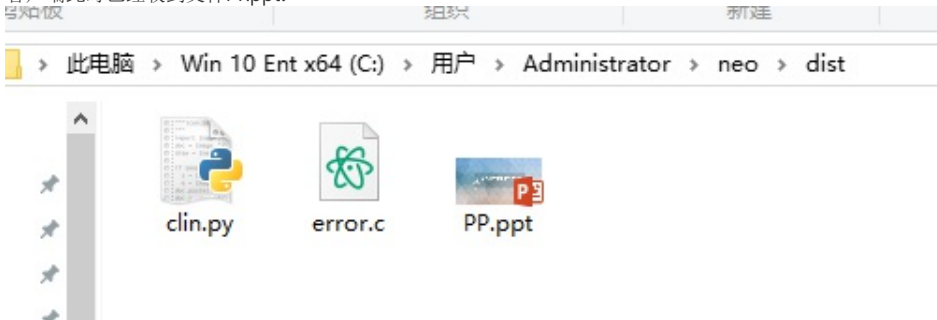
同时服务器端也能看到：

```

PP.ppt
Sending data...
Send success!
Cost 0.032000 seconds

```

客户端此时已经收到文件PP.ppt:



以上便是文件共享的全部功能，其实就是利用TCP进行数据传输的双向应用，只不过请求的发起人都是客户端。详细的程序说明请参看代码中的注释。

总结

本程序的核心功能在于对TCP连接的应用，以及多线程处理事件。通过这个程序，不仅了解了Python的多线程实现（实际上Python因为是用GIL实现的多线程，所以事实上是一种“假的”多线程，但是不妨碍理解多线程），更多的了解了多线程并发的不可预知性，如何将线程上锁，还有生产者-消费者的模式，让一些不可分割的操作捆绑成原子操作，以克服多线程的事件发生顺序不可预测的问题。

更深层次的了解TCP所谓的面向连接。在一次连接后，只要不断开，那么就可以不断地传送数据，至于数据是什么并不关键，他们只不过是一堆任计算机解释的0和1而已。

程序有很多不足之处，比如没有GUI，而且对用户不是十分友好，需要记忆命令行。另一方面，当服务器进行文件的接收和传送时，服务器运行的线程必须优先处理文件传输，此时如果有外界的信息传过来（比如另一个用户发了一句消息），那么该消息将阻塞，直到文件传输完成。这更像是单线程。但是我没有找到更好的办法，能够同时接收/传送并处理两个端口的数据又不至于混淆，这是我水平不足的地方，尚不能熟练地使用多线程。

总的来说，本程序除了实现了大体的功能以外，可以优化的地方还很多，即使如此也已经让我们受益良多。