CSC 573 – Internet Protocols
Project #1
Spring 2019

Rajat Narang(rnarang) and Udit Mishra(umisra)

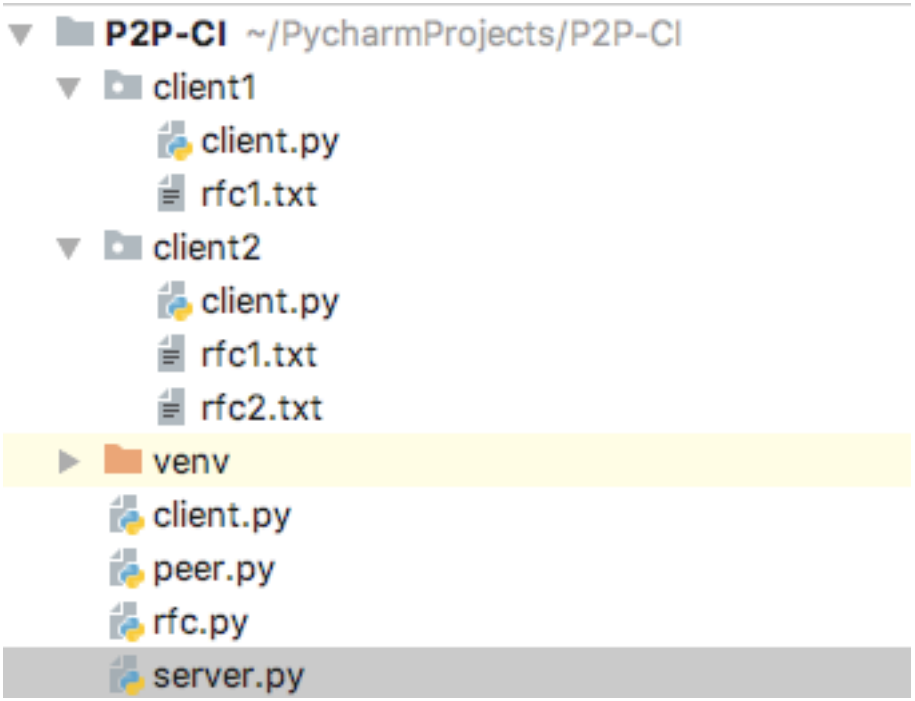
Steps to execute the program.

Requirements: Python 3.6+

For now the system runs on a local computer. To run on different computer, we need to change the server address in client.py on line 12

```
9
10      class Client:
11          def __init__(self):
12              self.server_IP = socket.gethostbyname('localhost')
13              self.server_port = 7734
14
```

Directory structure:



In order to support functionality for server, we have peers.py and rfc.py which are used to perform linked list operations.

- Open console and go to the project root directory. Run “python server.py”. You will see “Server started, listening for connections! ” on the console.

```
Rajats-Air:P2P-CI rajatnarang$ python server.py
Server started, listening for connections!
```

- Go to directory “client1” and run “python client.py”. You will be asked to connect to the server (y/n)?. Press “y” and client will be connected to server. You will see the following in your console.

```
Rajats-Air:client1 rajatnarang$ python client.py
Connect to Server, (y/n)? y
P2P-CI/1.0 200 OK
Hello!!
List of methods available:
1. ADD: add an RFC to the peer to peer network
2. LOOKUP: find peers that have a specified RFC
3. LIST: list all RFCs available
4. GET: download an RFC
5. EXIT: terminate connection
Select option - 1, 2, 3, 4 or 5
```

At the same time, on the server side:

```
Got connection from ('127.0.0.1', 49746)
```

- Currently, we don’t have any RFCs so we will add one. To add RFC, press 1.

  Client 1 has rfc1.txt present in its directory.

  Press 2 and add rfc number and rfc title.

  In our case rfc number=1 and rfc title=rfc1.txt

```
Select option - 1, 2, 3, 4 or 5
1
Enter RFC number: 1
Enter RFC title: rfc1.txt
P2P-CI/1.0 200 OK
1 rfc1.txt Rajats-Air@603 43225
```

  Response received from the server is printed in red color.

- Now, in another terminal window, go to root directory of project and open directory “client2” and run “python client.py”. After it runs, Press “y” to connect to server.

  Press 1 to add rfc. RFC number will be 2 and rfc title will be “rfc2.txt”.

```
Rajats-Air:client2 rajatnarang$ python client.py
Connect to Server, (y/n)? y
P2P-CI/1.0 200 OK
Hello!!
List of methods available:
1. ADD: add an RFC to the peer to peer network
2. LOOKUP: find peers that have a specified RFC
3. LIST: list all RFCs available
4. GET: download an RFC
5. EXIT: terminate connection
Select option - 1, 2, 3, 4 or 5
1
Enter RFC number: 2
Enter RFC title: rfc2.txt
P2P-CI/1.0 200 OK
2 rfc2.txt Rajats-Air@61 48215
```

Press 3 to see all list of RFC.

```
Select option - 1, 2, 3, 4 or 5
3
P2P-CI/1.0 200 OK
2 rfc2.txt Rajats-Air@61 48215
1 rfc1.txt Rajats-Air@603 43225
```

- Now we have two peers connected to the server. They can exchange files with each other.

```
P2P-CI/1.0 200 OK
2 rfc2 Rajats-Air@22 41362
RFC downloaded
```

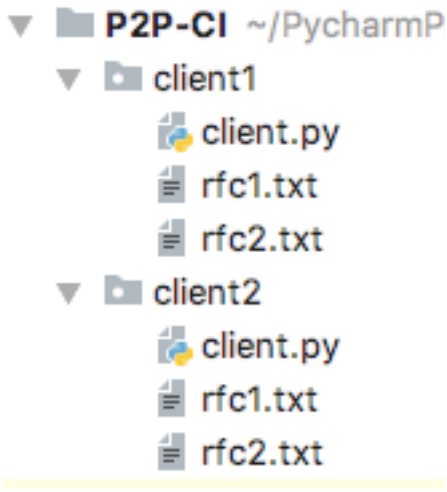
- Go to client 1 and press 5 to get the RFC. Insert rfc Number as 2. A file “rfc2” will be downloaded which can be seen in directory “client1”.
- Go to client 2 and Press 4 to download rfc1. Insert rfc number=1 to download rfc1.txt in client2.

  Once the file is downloaded you will see the response.

  After file is downloaded, Add RFC request is sent for the same file automatically.

```
Select option - 1, 2, 3, 4 or 5
P2P-CI/1.0 200 OK
1 rfc1 Rajats-Air@233 15036
RFC downloaded
P2P-CI/1.0 200 OK
1 rfc1 Rajats-Air@352 11887
```

- After successful transfer, we can check in the client1 and client2 directory. Both clients will have 2 files rfc1.txt and rfc2.txt.



- Client can terminate connection by pressing 5. Once the client connection terminated, it will be removed from the peer list and all related RFC will be removed from rfcList.