CSC 573 – Internet Protocols
Project #1
Fall 2020
Pranav Babulkar(pbabulk) and Shashank Joshi(sjoshi26)

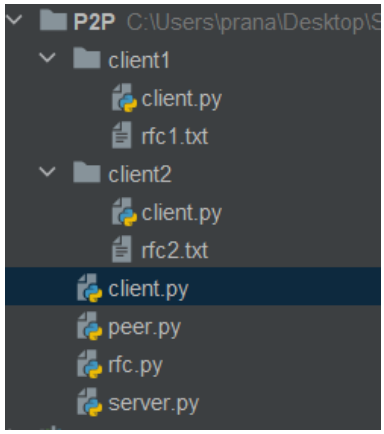
Steps to run the project:

Requirements: Python 3.6+

The project currently is configured to run on a local computer. For the peers to be separate machines the server address in client.py of each client can be changed on line 12.

```
1   import socket
2   import _thread
3   import os
4   from datetime import datetime
5   from pytz.reference import LocalTimezone
6   from platform import platform
7   from random import randint
8
9
10  class Client:
11      def __init__(self):
12          self.serverIP = socket.gethostbyname('localhost')
13          self.serverPort = 7734
```

Directory structure:



The directory structure is created as given in the image on the right. We have created 2 folders for 2 different clients inside which we are having separate rfc's for the peers to upload to the server. The client code is same for both the folders.
The files server.py acts as the server and listens to the connection requests and manages all the operations. It makes use of the 2 other files rfc.py and peer.py to manage the peers and rfcs in the server domain and make a linked list of those and then manage all the operations on those linked lists.

Executing the project:

1. **Starting the server:** Now to start the server, go to the root directory of the project and open console. Run “python server.py”. You will see a message like this as in the image below.

```
PS C:\Users\prana\Desktop\Sem 3\CSC 573\P1\P2P> python server.py
Server started, waiting for connections from Peers!
```

2. **Starting clients**: Now open 2 consoles and go to the client1 folder in one and client 2 folder in the other. Run “python client.py” on both. You will be prompted like this to connect to the server.

```
PS C:\Users\prana\Desktop\Sem 3\CSC 573\P1\P2P\client1> python client.py
Connect to Server, (y/n)?
```

3. Prompt ‘y’ on both the console windows to connect to the server. When prompted with ‘y’ you will see a Hello message and a list of commands you can use.

```
P2P-CI/1.0 200 OK
Hello, You are now connected to the server.
List of methods available:
1. ADD: Add an RFC in 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
```

Meanwhile on the server console you will be able to see 2 new connections registered when successfully connecting the peers to server.

```
PS C:\Users\prana\Desktop\Sem 3\CSC 573\P1\P2P> python server.py
Server started, waiting for connections from Peers!
Got connection from ('127.0.0.1', 61616)
Got connection from ('127.0.0.1', 61617)
```

4. Now that the peers are connected to the server. We can perform the operations like ADD(upload an RFC to the network), LIST(list all the RFC’s available in the network), LOOKUP(find peers that have a specific RFC), GET(download an RFC available) and terminate the connection from the server of a specific peer.
5. **Executing Operations:**
    a. **ADD:**
        i. From the client1(C1) and client2(C2) console press 1.
        ii. The console will prompt for the RFC number (Enter 1 for C1 and 2 for C2).
        iii. Now the program will prompt for RFC Title. (Enter ‘rfc1’ for C1 and ‘rfc2’ for C2). The output will be as follows.

```
Hello, You are now connected to the server.
List of methods available:
1. ADD: Add an RFC in 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: 1
Enter RFC title: rfc1
P2P-CI/1.0 200 OK
1 rfc1 G7@77 20420
```

C1 ADD Operation

```
Hello, You are now connected to the server.
List of methods available:
1. ADD: Add an RFC in 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
P2P-CI/1.0 200 OK
2 rfc2 G7@816 27477
```

C2 ADD operation

b. LIST:

i. Now that we have added 2 RFC's to the server domain. So when we enter 3 in either of the C1 or C2 console we will have both the RFC's visible as output.

```
Hello, You are now connected to the server.
List of methods available:
1. ADD: Add an RFC in 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
3
P2P-CI/1.0 200 OK
2 rfc2 G7@816 27477
1 rfc1 G7@77 20420
```

c. LOOKUP:

i. We can also lookup in the domain to determine which peers have a particular RFC using the LOOKUP option 2.

ii. You will be prompted to enter the RFC number and title which needs to be looked up. And we can see the output in red.

```
Hello, You are now connected to the server.
List of methods available:
1. ADD: Add an RFC in 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
2
Enter RFC number: 1
Enter RFC title: rfc1
P2P-CI/1.0 200 OK
1 rfc1 G7@77 20420
```

d. GET:

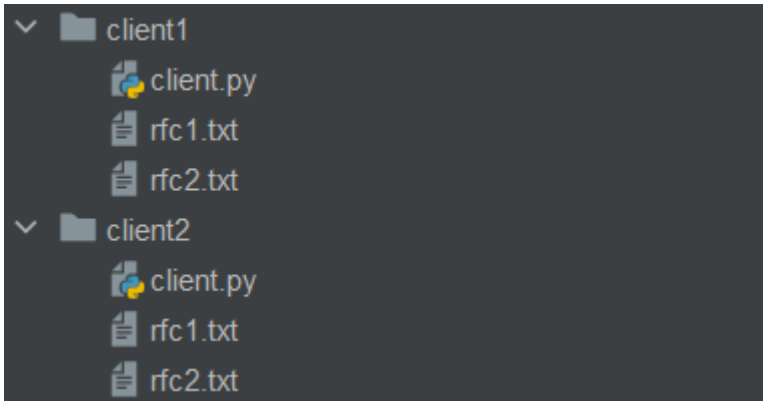
i. As the peers are connected to the server domain, so we can download the RFC's uploaded by the peers over the network to the server as long as they are connected to the server.

ii. Press 4 to download the RFC for the peers (C1 and C2). You will be prompted to enter the RFC number which needs to be downloaded and the output is shown as follows.

```
4
Enter RFC number: 1

Hello, You are now connected to the server.
List of methods available:
1. ADD: Add an RFC in 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
P2P-CI/1.0 200 OK
1 rfc1 G7@77 20420
RFC downloaded
P2P-CI/1.0 200 OK
1 rfc1 G7@816 27477
```

iii. After downloading the RFCs for C1 and C2 the RFC's are downloaded to the client's root directory and the directory structure looks something like in the screenshot below.



e. EXIT:

i. Pressing 5 will terminate the connection of the respective peer from the server and remove the RFC's belonging to that too.