

Test: Establish TCP connection

Input:

- Start tracker.py with command line arguments: 127.0.0.1 8080
- Start filesynchronizer.exe with command line arguments: 127.0.0.1 8080

Expected output: "Client connected"

Output:

```
C:\Users\drumm\Files\Coding\GitLab\salayog\PA2>tracker.py 127.0.0.1 8080
Waiting for connections on port 8080
Client connected with 127.0.0.1:56439
```

Result: Pass

Test: Synchronizing files

Input:

- test1 folder with filesynchronizer.exe and file1.txt
- test2 folder with filesynchronizer.exe and file2.txt
- Running tracker.py and both copies of filesynchronizer.exe with command line arguments: 127.0.0.1 8080

Expected output: After running the program, a copy of file1.txt is in the test2 folder and a copy of file2.txt is in the test1 folder

Output: Files copied successfully

Result: Pass

Test: Deserialize received JSON from peer

Input:

- test1 folder with filesynchronizer.exe and file1.txt
- Start tracker.py with command line arguments: 127.0.0.1 8080
- Start filesynchronizer.exe with command line arguments: 127.0.0.1 8080

Expected output: No JSONDecodeError thrown

Output: No errors thrown

Result: Pass

Test: Track new peer and file

Input:

- test1 folder with filesynchronizer.exe and file1.txt
- Start tracker.py with command line arguments: 127.0.0.1 8080
- Start filesynchronizer.exe with command line arguments: 127.0.0.1 8080

Expected output: New peer and file added to self.users and self.files dictionaries and printed

\*Temporarily added print statements for testing

Output:

```
C:\Users\drumm\Files\Coding\GitLab\salayog\PA2>tracker.py 127.0.0.1 8080
Waiting for connections on port 8080
Client connected with 127.0.0.1:50653
self.users:
('127.0.0.1', 8001) => 21.0232334
self.files:
file1.txt => {'ip': '127.0.0.1', 'mtime': 1552686850.1841009, 'port': 8001}
self.users:
('127.0.0.1', 8001) => 26.0464582
self.files:
file1.txt => {'ip': '127.0.0.1', 'mtime': 1552686850.1841009, 'port': 8001}
```

Result: Pass

Test: Peer and file already exist in dictionaries, peer version of file has new modified time, and test keepalive message

Input:

- test1 folder with filesynchronizer.exe and file1.txt
- Initialize self.users to contain ('127.0.0.1', 8001) : 0
- Initialize self.files to contain 'file1.txt' : {'ip': '127.0.0.1', 'mtime' : 9999999999999999, 'port' : 8001}
  - \*mtime is so large since it must be greater than the mtime received by the tracker
- Start tracker.py with command line arguments: 127.0.0.1 8080
- Start filesynchronizer.exe with command line arguments: 127.0.0.1 8080

Expected output:

- Peer and file are not added to self.users and self.files since they already exist
- mtime for already existing file is updated
- Keepalive message acknowledged and timer refreshed for that peer

Output:

```

C:\Users\drumm\Files\Coding\GitLab\salayog\PA2>tracker.py 127.0.0.1 8080
Waiting for connections on port 8080
Client connected with 127.0.0.1:60837
Peer already tracked!
File already tracked!
mtime updated
self.users:
('127.0.0.1', 8001) => 0
self.files:
file1.txt => {'ip': '127.0.0.1', 'mtime': 1552686850.1841009, 'port': 8001}
Keepalive refreshed
self.users:
('127.0.0.1', 8001) => 17.0052287
self.files:
file1.txt => {'ip': '127.0.0.1', 'mtime': 1552686850.1841009, 'port': 8001}

```

\*Print statements added for testing

Result: Pass

Test: Send directory response message

Input:

- test1 folder with filesynchronizer.exe and file1.txt
- Start tracker.py with command line arguments: 127.0.0.1 8080
- Start filesynchronizer.exe with command line arguments: 127.0.0.1 8080

Expected output: self.files sent through TCP connection

Output:

```

C:\Users\drumm\Files\Coding\GitLab\salayog\PA2>tracker.py 127.0.0.1 8080
Waiting for connections on port 8080
Client connected with 127.0.0.1:60883
Message sent:
{'file1.txt': {'ip': '127.0.0.1', 'mtime': 1552686850.1841009, 'port': 8001}}
Message sent:
{'file1.txt': {'ip': '127.0.0.1', 'mtime': 1552686850.1841009, 'port': 8001}}

```

\*Print statements added for testing

Result: Pass

Test: Timeout

Input:

- test1 folder with filesynchronizer.exe and file1.txt

- Start tracker.py with command line arguments: 127.0.0.1 8080
- Start filesynchronizer.exe with command line arguments: 127.0.0.1 8080
- filesynchronizer.exe closed after host and file1.txt have been added to self.users and self.files

Expected output: Tracker removes host and file1.txt from dictionaries after TTL has expired

Output:

```
C:\Users\drumm\Files\Coding\GitLab\salayog\PA2>tracker.py 127.0.0.1 8080
Waiting for connections on port 8080
Client connected with 127.0.0.1:56503
self.users:
('127.0.0.1', 8001) => 11.2150631
self.files:
file1.txt => {'ip': '127.0.0.1', 'mtime': 1552686850.1841009, 'port': 8001}
self.users:
('127.0.0.1', 8001) => 16.214891
self.files:
file1.txt => {'ip': '127.0.0.1', 'mtime': 1552686850.1841009, 'port': 8001}
Connection lost
self.users:
self.files:
Deleted!
```

\*Print statements added for testing

Result: Pass