Joshua Talbot, jpt3988@rit.edu
Ethan Nunez, ern1274@rit.edu

Peer-to-Peer Project

Tracker Output (python p2p_command -- tracker):

```
PS C:\Users\joshu\Downloads\CSCI 351 Project\P2P_FileShare_Project> python p2p_command.py --tracker
[Tracker] Running on 0.0.0.0:9000
[Tracker] Waiting for peers to connect...
[Tracker] Registered peer: 129.21.136.207:10001:Tempest
```

```
PS C:\Users\joshu\Downloads\CSCI 351 Project\P2P_FileShare_Project> python .\p2p_command.py
Enter port number (e.g., 10001): 10001
Enter peer name: Tempest
[System] Receiver thread launched.
[System] Exchange processor thread launched.
--- P2P File Sharing System ---
Hello, Tempest (listening on port 10001)

Current Peers:

Available commands:
i -peer_name          : Display data available from peer
c -peer_name -id      : Connect to peer and request file with id
r                     : Refresh peer discovery
q                     : Quit

Choose:
```

User input for ports and peer name:

```
PS C:\Users\joshu\Downloads\CSCI 351 Project\P2P_FileShare_Project> python .\p2p_command.py
Enter port number (e.g., 10001): 10001
Enter peer name: Tempest
[System] Receiver thread launched.
[System] Exchange processor thread launched.
--- P2P File Sharing System ---
Hello, Tempest (listening on port 10001)

Current Peers:

Available commands:
i -peer_name          : Display data available from peer
c -peer_name -id      : Connect to peer and request file with id
r                     : Refresh peer discovery
q                     : Quit

Choose:
```

Refreshing peer discovery:

```
Enter peer name: Tempest
[System] Receiver thread launched.
[System] Exchange processor thread launched.
--- P2P File Sharing System ---
Hello, Tempest (listening on port 10001)

Current Peers:

Available commands:
i -peer_name          : Display data available from peer
c -peer_name -id      : Connect to peer and request file with id
r                     : Refresh peer discovery
q                     : Quit

Choose: r

Current Peers:
 - Josh

Available commands:
i -peer_name          : Display data available from peer
c -peer_name -id      : Connect to peer and request file with id
r                     : Refresh peer discovery
q                     : Quit

Choose:
```

Displaying data available from a peer:

```
--- P2P File Sharing System ---
Hello, Josh (listening on port 10003)

Current Peers:
 - Tempest

Available commands:
i -peer_name          : Display data available from peer
c -peer_name -id      : Connect to peer and request file with id
r                     : Refresh peer discovery
q                     : Quit

Choose: i Tempest
[Index] Requesting index from ('129.21.136.207', 10001)...
[Remote Index]
ID: 001 -> shared/file1.txt
ID: 002 -> shared/file2.txt
ID: 003 -> shared/file3.txt
```

Connecting to peers and requesting file with ID:
Sender sends the request, receiver is rec

Sender Side:

```
Choose: c Tempest 001
[Exchange] Attempting to download file ID 001 from Tempest...
[Exchange] EXCH_REQ sent. Receiver will auto-save once the remote peer responds.

Current Peers:
 - Tempest

Available commands:
i -peer_name          : Display data available from peer
c -peer_name -id      : Connect to peer and request file with id
r                     : Refresh peer discovery
q                     : Quit

Choose: [Receiver] Received final chunk (-1) for file 001 -- writing to disk.
[Receiver] Saved file 001_torrent.txt successfully.
```

Receiver Side:

```
--- P2P File Sharing System ---
Hello, Tempest (listening on port 10001)

Current Peers:

Available commands:
i -peer_name          : Display data available from peer
c -peer_name -id      : Connect to peer and request file with id
r                     : Refresh peer discovery
q                     : Quit

Choose: r

Current Peers:
 - Josh

Available commands:
i -peer_name          : Display data available from peer
c -peer_name -id      : Connect to peer and request file with id
r                     : Refresh peer discovery
q                     : Quit

Choose: [Receiver] Received INDEX_REQ
[Receiver] Sent index response
[Receiver] Inserting EXCH_REQ with file id: 001 and peer_addr: 129.21.136.207:10003
[System] Fulfilling Exchange Request
[Sender] Starting file exchange for C:\Users\joshu\Downloads\CSCI 351 Project\P2P_FileShare_Project\shared\file1.txt
[Sender] Received ACK for packet 1
[Sender] Sent FIN, waiting for final ACK...
[Sender] Received final ACK. Transfer complete.
[Sender] Exiting FIN handshake sequence.
```

The user uses command-line interface to quit:

```
PS C:\Users\joshu\Downloads\CSCI 351 Project\P2P_FileShare_Project> python p2p_command.py
Enter port number (e.g., 10001): 10008
Enter peer name: Potato
[System] Receiver thread launched.
[System] Exchange processor thread launched.
--- P2P File Sharing System ---
Hello, Potato (listening on port 10008)

Current Peers:

Available commands:
i -peer_name          : Display data available from peer
c -peer_name -id      : Connect to peer and request file with id
r                     : Refresh peer discovery
q                     : Quit

Choose: q
Leaving system. Goodbye!
PS C:\Users\joshu\Downloads\CSCI 351 Project\P2P_FileShare_Project>
```

Design Decisions:

Used a central tracker server with which peers register with.

Peers send direct EXCH_REQ messages containing the file ID and IP address.

Files that are available to share in between peers are unfortunately hard-coded and inside the “shared” folder for this project.

One peer sends the entire file to another peer, chunk by chunk. No splitting of work among multiple sources.

Each peer runs a receiver thread and a file-sending processor thread.