Bittorrentpy

1.Protocol

There are 4 types of messages in the **Tracker protocol**. Each of them is implemented in Tracker class. You can check the class for details.

```
class MessageTypes(Enum):
        GET_FILE_LIST = 1
        GET_CHUNK_LOCATIONS = 2
        FILE_REGISTER = 3
        CHUNK_REGISTER = 4
```

An example communication:

1. Client send GET_CHUNK_LOCATIONS message containing the payload “filename”
2. Server gets the metadata for that file and sends the size of the next outgoing message
3. Then the server sends the metadata, a list of chunk hashes and peers that have that chunk.

Others message types follow a similar pattern and the flow can be found in network.py Tracker.handle_message.

There is 1 type of message in the **Peer protocol**.

1. Requester sends filename and chunk_id.
2. Provider sends the data.

2.Structure

The system has 3 modules.

1. Start.py: Gets user input and has the UI logic.
2. Network.py: Handles communication to Peers and the Tracker. Chunk_size is a constant.
3. Transfer.py: Downloads chunks concurrently. Concurrency_count is a constant.

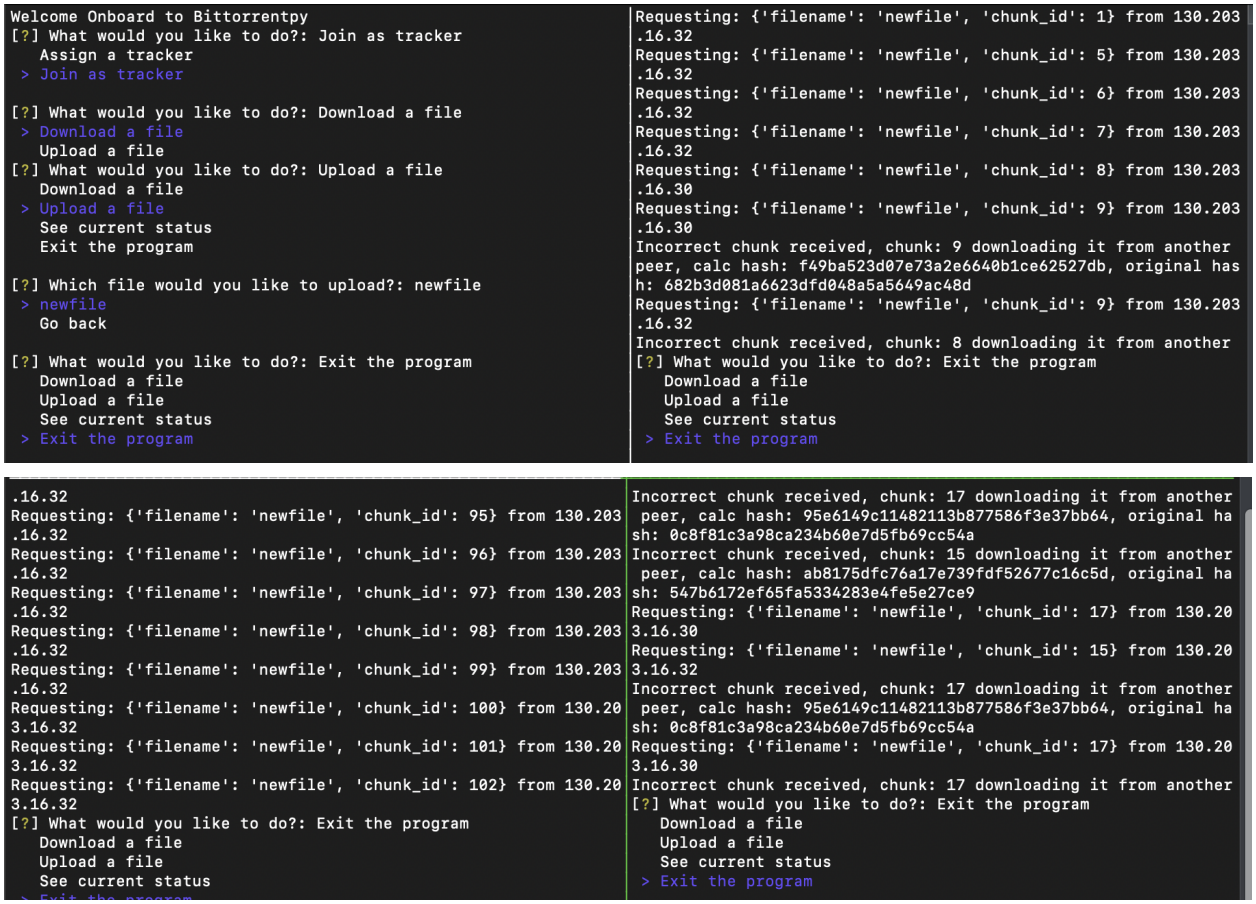
Files are downloaded into uploaded from the /files folder.

3.What Works

Everything works

4.Sample output

Example of integrity check:



5.Source code

Is provided inside the zip.

6.Run

The program is implemented with python3. The requirements are provided inside requirements.txt. To run the program use
$ python3 src/start.py

To run it with verbose on
$ python3 src/start.py v