SHRUTHI KATAPALLY
User Name: shrukata

Project 2: BitTorrent Client Implementation

INTRODUCTION:
This project gives the implementation of bittorrent client using basic socket programming with 1 Seeder –N Leecher. The client here requests the server the pieces of file it needs and will receive them if the server has the pieces of file. All the communication is done using a torrent file and piece number. This project submission has the following files:

- bt_client.c
- bt_lib.c
- bt_setup.c
- bt_lib.h
- bt_setup.h
- readme
- makefile

The project runs in two different modes:
1) Leecher mode: In Leecher mode, a socket will be created and it can be connected with a seeder giving its IP address and port where seeder is listening. If it gets connected, leecher starts a handshake where it sends the data-19Bittorrent Protocol,8 zero bytes for future use, info_hash, peer_id. The client now gets a response handshake if the seeder has the matched values. This has a bit field of the file leecher requested. Now Client will decide what all pieces it needs and sends the interested message to seeder and the data it needs. This continues till leecher has all the file and is no more interested in data receiving.
2) Seeder mode: In the seeder mode, a socket is created and it starts listening for incoming connections. Seeder after connecting to a client gets a handshake message from client which has the data -19Bittorrent Protocol, 8 zero bytes for future use, info_hash, peer_id. Seeder will calculate the hash values and will respond with a handshake if the values match. Seeder here sends the bit field of the file it has. Seeder will get a interested message and bitfield response if the client needs any data which seeder has. This continues till seeder receives the interested message.

Files in the project:

1) bt_client.c: This file has the parsing of torrent file, handshake, server and client functions where we send and receive the bits of files.
   void launchClient(bt_args_t * bt_args);
   void launchServer(bt_args_t * bt_args);
   void parse_torrent(char * torrent_file);
2) bt_setup.c: This file has functions for parsing the command line arguments and parsing the peer to store the data into peer_t structure.
   void __parse_peer(peer_t * peer, char * peer_st);`
   void parse_args(bt_args_t * bt_args, int argc, char * argv[]);

3)bt_lib.c: This file has functions to calculates the peer id and also assigns the peer data of ip and port number.

SHRUTHI KATAPALLY
User Name: shrukata

Project 2: BitTorrent Client Implementation

```
void calc_id(char * ip, unsigned short port, char *id);
int init_peer(peer_t *peer, char * id, char * ip, unsigned short port);
```

The Makefile has the following :
A **Makefile** is written for the ease of compilation. The files bt_client.c, bt_setup.c,bt_lib.c are complied and an object file bt_client.o is generated using this makefile. It is also used for removing the previously generated object files.

The README has the following :
A **README** gives the basic idea of the implementation of project which is used in running the project. It will also give the procedure to execute the program. It also gives how to use the Makefile to compile the program. Readme also has the output analyzation.
References and Credits:

Project Partners:
Jagadeesh Madagundi (jmadagun)
Om Harshini Sowmya Achanta (omachant)

Books and references:

1) TCP/IP Sockets in C: A Practical Guide for Programmers 2
2) https://wiki.theory.org/BitTorrentSpecification#Bencoding
3) https://computing.llnl.gov/tutorials/pthreads/#Pthread
4) http://ahvaz.ist.unomaha.edu/azad/temp/bittorrent/05-fonseca-bittorrent-protocol.pdf