**README**

In a simple client/server paradigm network, usually a very large file is chunked into n small files, i.e. chunk 1 to chunk n, that are shared among the clients. In this project a client is a user who wants information about other clients who own the chunk required by the client. In turn, when asked, a client can also upload some chunks that he has to other clients. In order to let other clients know that he has these chunks, a centralized server is utilized to keep track of the chunks status. In this project, there are 64 chunks. Each client maintains a bit vector, called “chunk vector” to indicate that whether he has this chunk or not. If he has this chunk, the bit is set 1. Otherwise, the bit is set to 0.

**Server**

1).The server has 3 files: server.h, server.c , functions.c

2). Server.c contains the main function, functions.c has functions definitions

3). To build the server, go to directory server and run

$cd server

$ gcc –o server.c functions.c

4). To run the server

$ ./server

5). To quit the server type

$ q

**Client**

1).The client folder has 2 files and 1 directory: client.h, client.c, config.

2).client.c contains the main function and required function definitions. The config folder has all the configuration files.

3). To build the client, go to directory client and run

$cd client

$ gcc –o client.c

4). To run the client

$ ./client SERVER\_IP Configuration\_file

5). To quit the client type

$ q