**Client Functionalities**

In this project a client is created who maintains a bit vector, called chunk vector to indicate that whether he has this chunk or not. For this project, there are 64 chunks. If he has this chunk, the bit is set 1. Otherwise, the bit is set to 0. 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 turn, when asked, a client can also upload some chunks that he has to other clients.

The client will be invoked with a command line which contains two arguments. The first argument is either the server’s IP address or the server’s domain name. The second argument is the name of a configuration file that he can read information from.

Function declarations

**int get\_config(char \*, int \*, char \*, char \*)**

- The user provides a configuration file from where the client gets details about server port, file vector, client id and client listening port.

**void read\_from\_activesockets(void)**

- Read data from incoming active ports. The port can also be stdin, for the client to enter requests from the keyboard.

**void read\_config();**

- The client reads the details from configuration file provided by the user.

**int connect\_to\_server();**

-Connect with the server to register and get details of other client who owns the chunk.

**void process\_stdin\_message();**

- Process the input from stdin. If the use enters ‘f’ ask user which chunk is required. If the user enters ‘q’ sends Quit message to server and wait till server sends ‘Exit’, then close the connection.

**void client\_run();**

- Execute client functionalities once its connected to the server.