The server is started by myserver <portnumber>

Server is a single threaded program . It loops reading from the udp socket and proceses eash message. If the message is valid client request the server tries to return proper response.
The server quits if it encounter any error but the file related and commnication with client related ones. If it encounters conditions like missing file, cannot fopen or cannot seek it reponds back to the client the error condition. Server doesnt handle any input and it must be killed to be closed.

The client is started by myclient <servers text file> <number of chunks>. Then it reads the list of servers, up to a maximum of **MAX_SERVERS** defined as 20 and wait for input of filename. Dwn_ is prepended to the output filename.

The client tries to connect to some server to get the file size, it iterates over all servers in order. If no server responds with valid filesize the client thinks it cannot download the file and fails 'Cannot transfer this file'.

If a correct file size is receiver the client divides the filesize into **numchunks** chunks, allocates parameter block for each chukn (containg the offset, chunksize, file pointer, sinchronisation mutex and a few other fields) and spawns **numchunks** threads.

Each thread tries t ocommunicate with the initial server, if the communication fails (either at the beggingin ,or dirring the file transfer) it switches to the next server The thread sends packets to read consecutive parts of the file, of 1024 bytes maximum size, after all the data that the thread is responsible for writing is written the thread sets a success flag.

If after all threads terminate if some part is not successfully downloaded the client prints 'Download failed', else it prints 'Successfull download of filename'.