a. Full name: Bingxin Yan

b. Student ID: 7817662107

c. In this assignment, I have done 3 file servers, 2 clients and one directory server. All of these 3 kinds of servers/clients can communicate with each other, except some minor problems. In each communication, I created a link between “client” and “server” by using socket. After send or receive messages, I use the message to do some analysis. In this process, I used C to open a file, write the file, read the file and then use the information provided by the file.

d. directory\_server.c: communicate with file server and client. It gets the information from file server and store it in directory.txt. After that, the client can get the information of file server from it. It looks up resource.txt to find which file server(s) has(have) the requested document. If there are more than 1 file servers, it refers to topology.txt to choose a closest one and send the TCP port number of that file server to the client.

file\_server1.c: At the beginning, it registers with the directory server, which includes the server number and its TCP port number. Then it can provide the requested file to a specific client.

file\_server2.c: At the beginning, it registers with the directory server, which includes the server number and its TCP port number. Then it can provide the requested file to a specific client.

file\_server3.c: At the beginning, it registers with the directory server, which includes the server number and its TCP port number. Then it can provide the requested file to a specific client.

client1.c: send requests to file servers. Before that, it needs to ask directory server which file server can give it the required file.

client2.c: send requests to file servers. Before that, it needs to ask directory server which file server can give it the required file.

e. The orders of run each of these files is: directory, file\_server1, file\_server2, file\_server3, client1, client2. The order of file\_server1, file\_server2, file\_server3 can be changed and the order of client1 and client2 also can be changed. The directory server can send the information of file server that has the requested doc. But there are some problems with recvfrom function. So I use a fixed message here, which stored in an array. Actually, this client can handle the received message dynamically. Because of that, if you want to change some parts in the .txt files, you need to change the contents in the array. Before the end of phase 2, all things can work successfully. This means that even change the resource.txt and topology.txt, the result is correct before the last step of phase 2.

f. The format of the messages exchanged is the same as mentioned in the requirement file(i.e. EE450 Socket Programming Project Fall 2013\_Revised\_Last).

g. The directory server can send the information of file server that has the requested doc. But there are some problems with recvfrom function. So I use a fixed message here. Actually, this client can handle the received message dynamically.

h. I used some codes from Beej's Guide to Network Programming. The function I used are: socket(), bind(), connect(), listen(), accept(), send(), recv(), sendto(), recvfrom(), close(), getaddrinfo().