Sockets and Signals- Programming assignment

Instructions:

Modify the program described in Assignment 3 in such a way that the two programs can be run on different machines. The program receiver.c will now be the client program that will interact with the user. Call it client.c. The processor.c will be the backend program – call it server.c. The client and server programs have the same functionality as described in Assignment 3 except that they should use sockets as a communication mechanism that will allow the client and the server to be run on different machines.

When the user presses Ctrl-C on the server.c terminal, the server program should display a summary, which should show the number of lines received so far and the total digit count across all input seen by the server.

Steps:

The basic idea is that server.c running at back and waiting for client.c connection. And in client.c it will create a socket connection with the ip of 127.0.0.1, get input from user , check secrets code and then send to server. In server.c, it will calculate digits and lines after receiving data. When detect CTRL+c, it will print all information.

For client.c

1. define socket and sockaddr_in
2. connect to server

   connect(sock_client, (struct sockaddr *)&servaddr, sizeof(servaddr)
3. wait for user's input and send it to server

   while (fgets(sendbuf, sizeof(sendbuf), stdin) != NULL)

   send(sock_client, sendbuf, strlen(sendbuf),0)

For server.c

1. define socket and sockaddr_in
2. bind to client

   bind(server_sockfd,(struct sockaddr *)&server_sockaddr,sizeof(server_sockaddr))
3. listening data

   listen(server_sockfd,100)

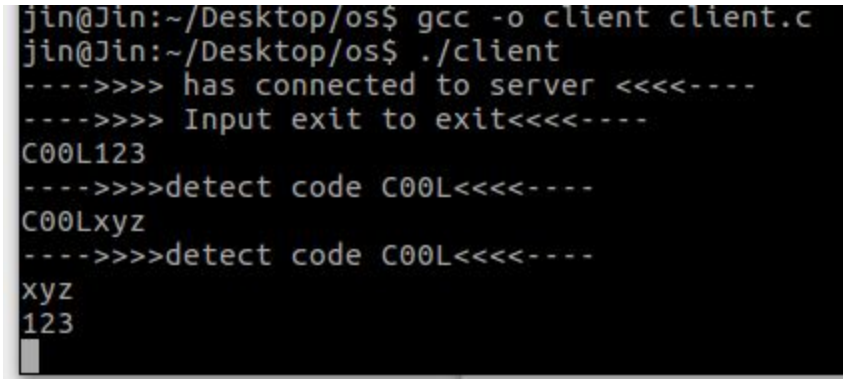
4. receive data and put it into buffer

recv(conn, buffer, sizeof(buffer),0)

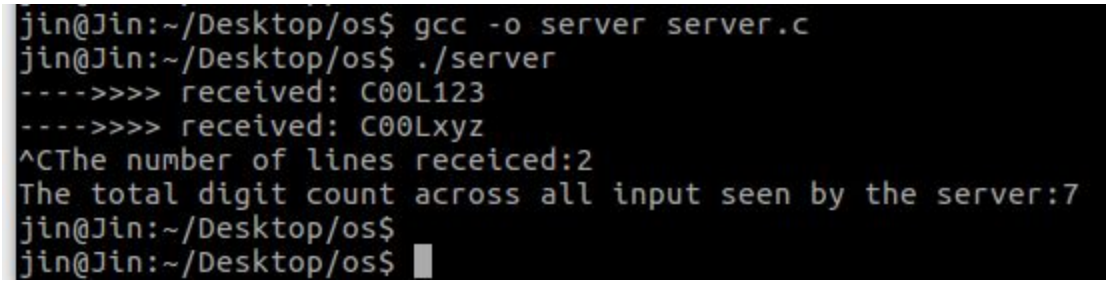
5. calculate digit , lines and write into secrets.
6. detect CTRL+C signal and print all information

Result:

client.c



Server.c



Secrets.out