

Name : Sabarivasan V

Class : CSE - B

Reg No : 205001085

Objective :

To develop an application for simple client server connection. Client sends a message to the server and echo back the same message to the client.

Server code :

```
#include <netinet/in.h> //structure for storing address information
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <sys/socket.h> //for socket APIs
```

```
#include <sys/types.h>
```

```
int main(int argc, char const* argv[])
```

```
{
```

```
    // create server socket similar to what was done in
```

```
    // client program
```

```
    int servSockD = socket(AF_INET, SOCK_STREAM, 0);
```

```
// string store data to send to client
//char message[20];
char serMsg[255] ;
printf("Enter message for client : ");
scanf("%s",serMsg);

//strcpy();

// define server address
struct sockaddr_in servAddr;

servAddr.sin_family = AF_INET;
servAddr.sin_port = htons(9001);
servAddr.sin_addr.s_addr = INADDR_ANY;

// bind socket to the specified IP and port
bind(servSockD, (struct sockaddr*)&servAddr,
    sizeof(servAddr));

// listen for connections
```

```
listen(servSockD, 1);

// integer to hold client socket.
int clientSocket = accept(servSockD, NULL, NULL);

// send's messages to client socket
send(clientSocket, serMsg, sizeof(serMsg), 0);
printf("Message: %s\n", serMsg);

return 0;
}
```

Client Code :

```
#include <netinet/in.h> //structure for storing address information
#include <stdio.h>
#include <stdlib.h>
#include <sys/socket.h> //for socket APIs
#include <sys/types.h>

int main(int argc, char const* argv[])
{
```

```
int sockD = socket(AF_INET, SOCK_STREAM, 0);
```

```
struct sockaddr_in servAddr;
```

```
servAddr.sin_family = AF_INET;
```

```
servAddr.sin_port
```

```
    = htons(9001); // use some unused port number
```

```
servAddr.sin_addr.s_addr = INADDR_ANY;
```

```
int connectStatus
```

```
    = connect(sockD, (struct sockaddr*)&servAddr,  
              sizeof(servAddr));
```

```
if (connectStatus == -1) {
```

```
    printf("Error...\n");
```

```
}
```

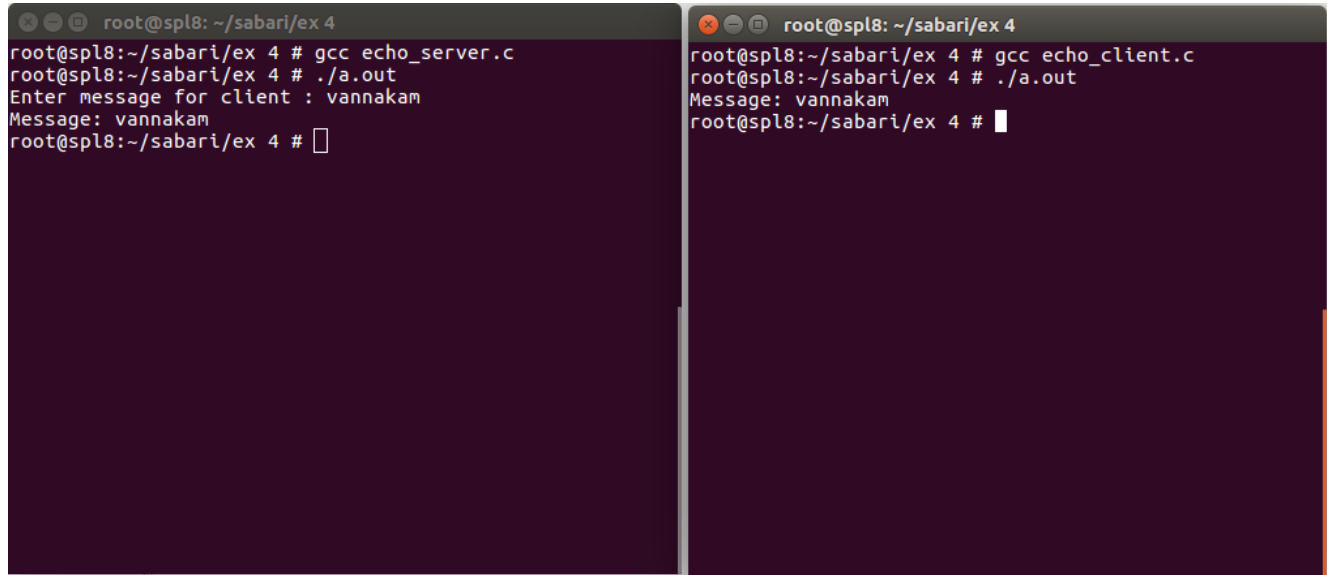
```
else {
```

```
    char strData[255];
```

```
    recv(sockD, strData, sizeof(strData), 0);
```

```
    printf("Message: %s\n", strData);  
}  
  
return 0;  
}
```

Output :



```
root@spl8: ~/sabari/ex 4  
root@spl8:~/sabari/ex 4 # gcc echo_server.c  
root@spl8:~/sabari/ex 4 # ./a.out  
Enter message for client : vannakam  
Message: vannakam  
root@spl8:~/sabari/ex 4 #  
  
root@spl8: ~/sabari/ex 4  
root@spl8:~/sabari/ex 4 # gcc echo_client.c  
root@spl8:~/sabari/ex 4 # ./a.out  
Message: vannakam  
root@spl8:~/sabari/ex 4 #
```