SSN COLLEGE OF ENGINEERING, KALAVAKKAM
(An Autonomous Institution, Affiliated to Anna University, Chennai)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NETWORKS LAB
EXERCISE 2

Name: Jayannthan P T *Dept: CSE 'A'* *Roll No.: 205001049*

Aim:

To transfer a file from server to client using TCP socket programming.

Algorithm:

CLIENT

1. Create a socket using socket() system call.
2. Connect it to the server.
3. Prompt the user to enter the file name.
4. Transfer the file name to the server
5. Receive the contents of the file and save in a new location
6. Close the socket

SERVER

1. Create a socket using socket() system call.
2. Bind the created socket with the port.
3. Listen for the connections.
4. When the server receives file name from the client, read the contents and send the contents to client.

Code:

Server

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

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

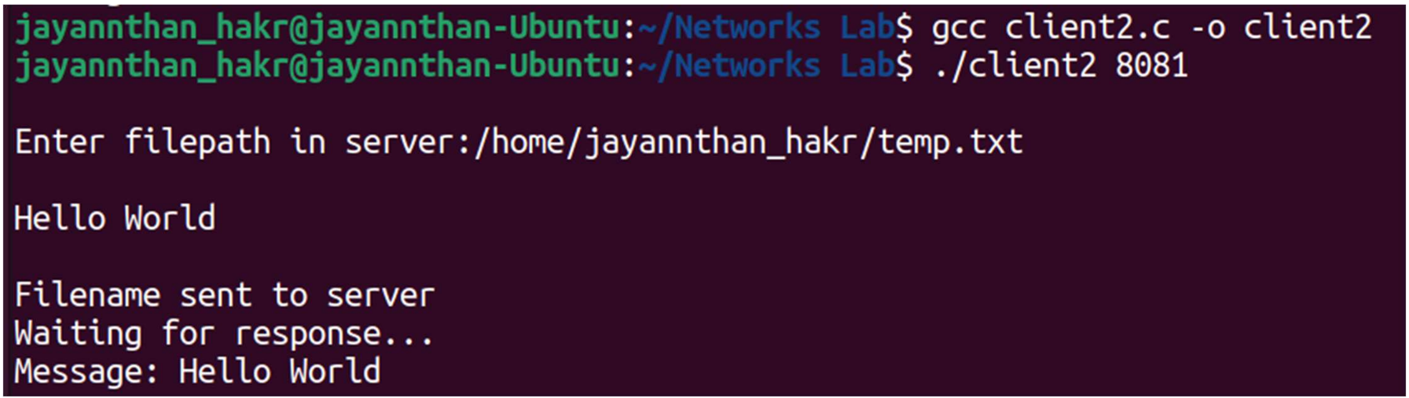
```
{
    int port=atoi(argv[1]);
    int serSockID = socket(AF_INET, SOCK_STREAM, 0);
    char fileText[255],filename[100];
    struct sockaddr_in servAddr;
    servAddr.sin_family = AF_INET;
    servAddr.sin_port = htons(port);
    servAddr.sin_addr.s_addr = INADDR_ANY;
    bind(serSockID, (struct sockaddr*)&servAddr,sizeof(servAddr));
    listen(serSockID, 1);
    int clientSocket = accept(serSockID, NULL, NULL);
    printf("\nConnected\n");
    int connected=1;
    /*while(connected)
    {*/
        read(clientSocket,filename,sizeof(filename));
        int fd=open(filename,O_RDONLY);
        read(fd, fileText, sizeof(fileText));
        printf("File Path Recieved: %s\nMessage read:%s\n!!!Sending the file to 
client!!!\n\n",filename,fileText);
        send(clientSocket, fileText, sizeof(fileText), 0);
        /*if(strcmp(serMsg,"exit")==0)
        {
            connected =0;
        }*/
    //}
    return 0;
}
```

Client

```
#include <netinet/in.h> //structure for storing address information
#include <stdio.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <string.h>
#include <sys/types.h>
#include <unistd.h>
#include<fcntl.h>
int main(int argc, char const* argv[])
{
    int port=atoi(argv[1]);
    int sockD = socket(AF_INET, SOCK_STREAM, 0);
    struct sockaddr_in servAddr;
    servAddr.sin_family = AF_INET;
    servAddr.sin_port= htons(port);
    servAddr.sin_addr.s_addr = INADDR_ANY;
    int connectStatus= connect(sockD, (struct sockaddr*)&servAddr,sizeof(servAddr));
    //printf("\nConnected\n");
    if (connectStatus == -1) {
        printf("Error...\n");
    }
    else
    {
        int connected=1;
        char fileText[255],filename[100];
        /*while(connected)
        {*/

            printf("\nEnter filepath in server:");
            scanf(" %s",filename);
            int fd=open(filename,O_RDONLY);
            read(fd, fileText, sizeof(fileText));
            printf("\n%s",fileText);
            int wr=write(sockD,filename,sizeof(filename));
            printf("\nFilename sent to server\nWaiting for response...\n");
            recv(sockD, fileText, sizeof(fileText), 0);
            printf("Message: %s\n", fileText);
            /*if(strcmp(strData,"exit")==0)
            {
                connected =0;
            }*/
    //  }
    }
    return 0;
}
```

Output:

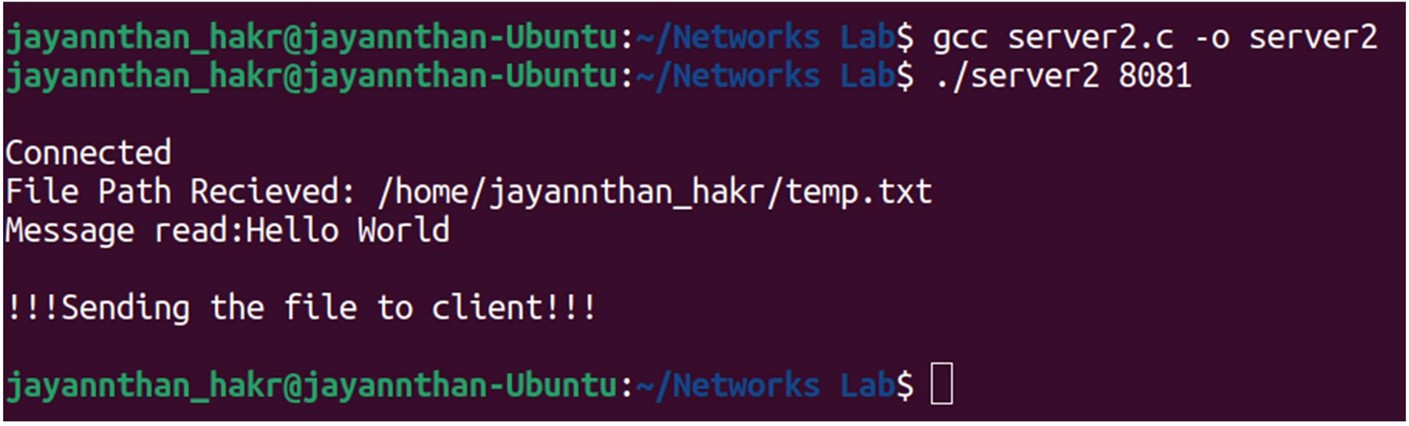


```
jayannthan_hakr@jayannthan-Ubuntu:~/Networks Lab$ gcc client2.c -o client2
jayannthan_hakr@jayannthan-Ubuntu:~/Networks Lab$ ./client2 8081

Enter filepath in server:/home/jayannthan_hakr/temp.txt

Hello World

Filename sent to server
Waiting for response...
Message: Hello World
```



```
jayannthan_hakr@jayannthan-Ubuntu:~/Networks Lab$ gcc server2.c -o server2
jayannthan_hakr@jayannthan-Ubuntu:~/Networks Lab$ ./server2 8081

Connected
File Path Recieved: /home/jayannthan_hakr/temp.txt
Message read:Hello World

!!!Sending the file to client!!!

jayannthan_hakr@jayannthan-Ubuntu:~/Networks Lab$ 
```

Learning outcome:

Learnt to create connection using sockets

Learnt to communicate between server and client using socket