Assignment 2

APPROACH: This a simple socket application which sends a file data from client and send to a server and server created a new file its directory and sends a confirmation to client again that file it received.

Open, read and write system calls are used for file input and output and socket communication is used to send file and get confirmation.

Server Side:

1. For socked I am using localhost "127.0.0.1" on port no: 8002 and IPV4 address is used in the code for IPV4 connection.
2. Write_file() functions receives the data from the client side and write it to the another file named a.txt on server side and also send a confirmation to the client the file has be written successfully.
3. The main function creates the socket enable it to listen to 10 connections at a time.

Client Side:

1. Similarly here the main function create and a socket and send a connection request to the server.
2. To integrate a task of using system my code is creating a file if doesn't exists and if exists then writing to that file on client using a write write_f() function, read that file using read system call and the ask the user for confirmation that he really wants to send that file to the server and if yes then send this file to server otherwise give an error.
3. And the send_to_file function take file pointer and socket information as an input and read the file and send it on that socket.

Server code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <arpa/inet.h>
```

```
#include<fcntl.h>
#define MAX_SIZE 1024 // maximum size of the buffer we are using


// function to write file data received on socket to file a.txt
void write_file_data(int sockfd){
  int n;
  FILE *fp;
  char *hello = "Got your file";
  char *filename = "a.txt";
  char buffer[MAX_SIZE];

  fp = fopen(filename, "w");
  while (1) {
    n = recv(sockfd, buffer, MAX_SIZE, 0); //recieve all data on buffer

    if (n <= 0){
      break;
      return;
    }
    fprintf(fp, "%s", buffer); //write received data to a file
    send(sockfd , hello , strlen(hello) , 0 );
    bzero(buffer, MAX_SIZE); //flush the buffer by writing all zeros till size
of buffer
  }
}

int main()
{
  char *ip = "127.0.0.1";
  int port = 8002;
  int s;
  //char *hello = "Got your file";

  int sockfd, new_sock;
```

```
 struct sockaddr_in server_addr, new_addr;
 socklen_t addr_size;
 char buffer[MAX_SIZE];

 sockfd = socket(AF_INET, SOCK_STREAM, 0); //intialize socket
 if(sockfd < 0) {
   perror("Error while creating socket");
   exit(1);
 }
 printf("Server socket created \n");

 //filling details for connection in structure of sockaddr_in
 server_addr.sin_family = AF_INET;  //intialize IPV4 family
 server_addr.sin_port = port; //provide port no
 server_addr.sin_addr.s_addr = inet_addr(ip); //providing IPV4

 s = bind(sockfd, (struct sockaddr*)&server_addr, sizeof(server_addr));
 if(s < 0) {
   perror("Error while binding");
   exit(1);
 }
 printf("Binded.\n");

 if(listen(sockfd, 10) == 0){
printf("Waiting for connection...\n");
}else{
perror("Error while listening");
   exit(1);
}

 addr_size = sizeof(new_addr);
 new_sock = accept(sockfd, (struct sockaddr*)&new_addr, &addr_size);
//accept reuested connection

 write_file_data(new_sock); //call our write_file_data function
```

```
  printf("Data written in the file successfully.\n");
 // send(new_sock , hello , strlen(hello) , 0 );



  return 0;
}
```

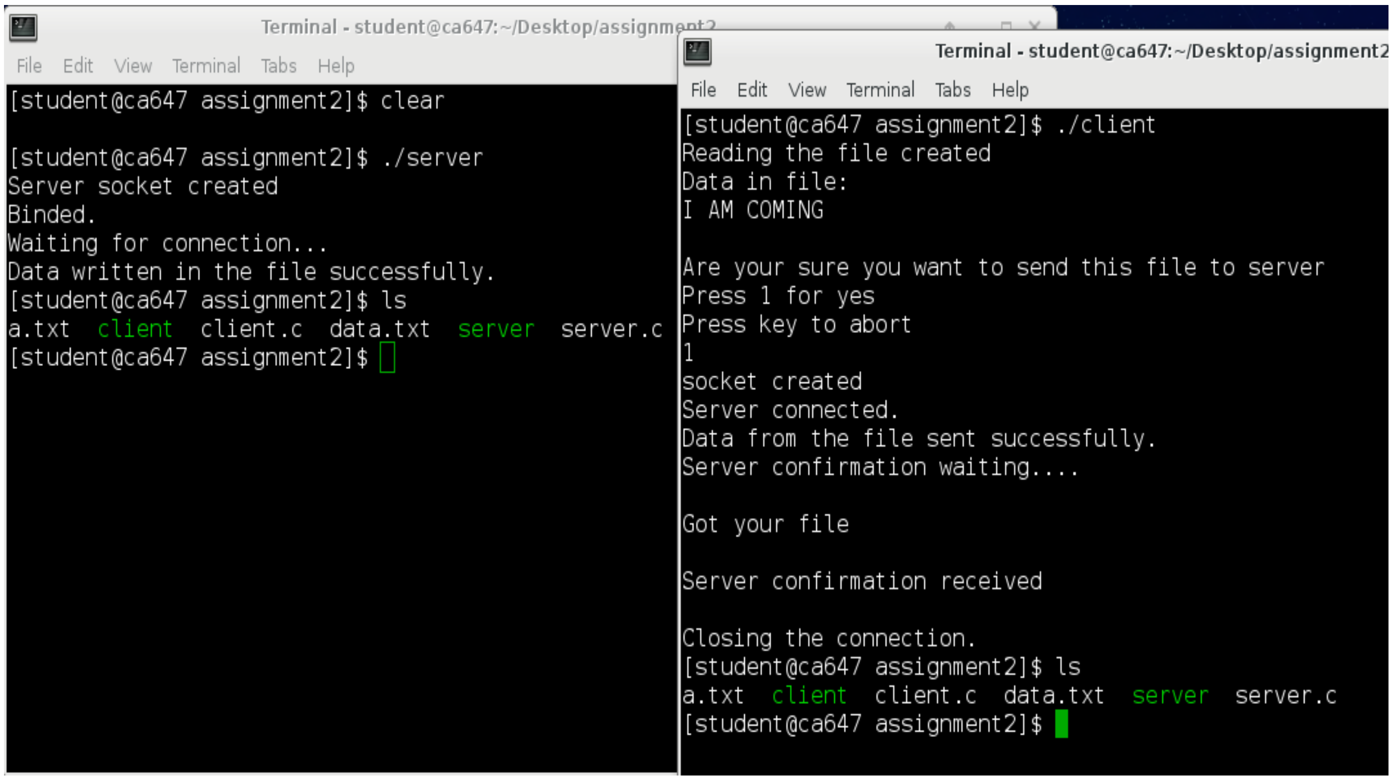
Client code

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <arpa/inet.h>
#include<fcntl.h>
#define MAX_SIZE 1024

//function to send file data on socket
void send_to_file(FILE *fp, int sockfd){
  int n;
  char data[MAX_SIZE] = {0};

  while(fgets(data, MAX_SIZE, fp) != NULL) {
    if (send(sockfd, data, sizeof(data), 0) == -1) {
      perror("ERROR WHILE SENDING FILE");
      exit(1);
    }
    bzero(data, MAX_SIZE);
  }
}

// function to create and write our data to a file
int write_f()
{
        int f_id;
        f_id = open("./data.txt", O_WRONLY);
        write(f_id, "I AM COMING\n", strlen("I AM COMING\n"));
        close(f_id);
        return 0;
}

//function to read the data from the file created
int read_f()
{
  int f_id;
  char *buffer = (char *) calloc(100, sizeof(char));
  f_id = open("./data.txt", O_RDONLY);
  int bytes = read(f_id, buffer, 100);
  printf("Reading the file created\n");
  printf("Data in file:\n%s\n",buffer);
}

int main(){
  char *ip = "127.0.0.1";
  int port = 8002;
  int s;

  //creating a socket
  int sockfd,valread;
  struct sockaddr_in server_addr;
  FILE *fp;
  char buffer[1024] = {0};
  write_f(); //calling write_f function
  read_f();  // calling read_f function
  int option; //option for confirmation
  char *filename = "data.txt"; //read this file and send data to server side

  printf("Are your sure you want to send this file to server\nPress 1 for yes\
nPress key to abort\n");
```

```
 scanf("%d",&option);

 //Ask for confirmation
 if(option == 1){
 sockfd = socket(AF_INET, SOCK_STREAM, 0);
 if(sockfd < 0) {        //through error if some error with socket creation
   perror("Error while crerating socket");
   exit(1);
 }
 printf("socket created\n");

 server_addr.sin_family = AF_INET; //define family for IPV4
 server_addr.sin_port = port; //our connection port no
 server_addr.sin_addr.s_addr = inet_addr(ip); //IP for connection

 s = connect(sockfd, (struct sockaddr*)&server_addr, sizeof(server_addr));
 if(s == -1) {
   perror("Error in socket");
   exit(1);
 }
printf("Server connected.\n");

 fp = fopen(filename, "r"); //open a file
 if (fp == NULL) {
   perror("Error while reading file.");
   exit(1);
 }

 send_to_file(fp, sockfd); //send file pointer to server
 printf("Data from the file sent successfully.\n");
 printf("Server confirmation waiting....\n\n");
 valread = read(sockfd , buffer, 1024);
 printf("%s\n\n",buffer );
 printf("Server confirmation received\n\n");
```

```
  printf("Closing the connection.\n");
  close(sockfd);}

  //generate error if user denied to send file
  else{
  printf("File Aborted");
  }

  return 0;
}
```

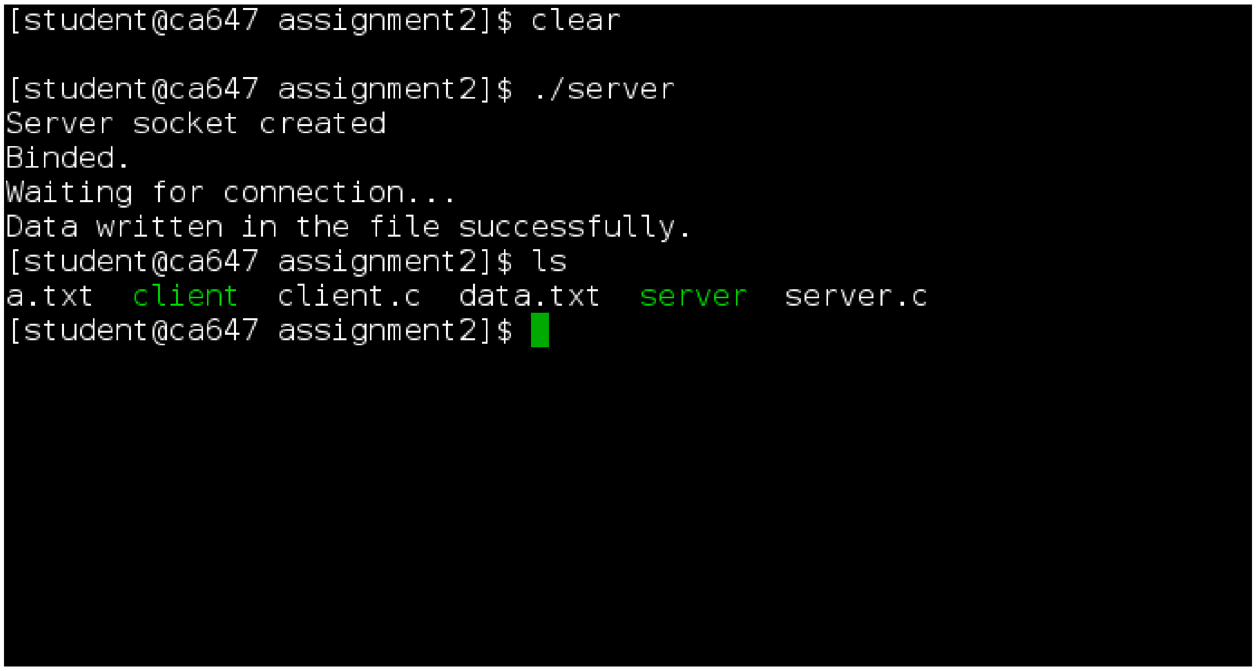
OUTPUT SCREENSHOOT



```
[student@ca647 assignment2]$ clear

[student@ca647 assignment2]$ ./server
Server socket created
Binded.
Waiting for connection...
Data written in the file successfully.
[student@ca647 assignment2]$ ls
a.txt  client  client.c  data.txt  server  server.c
[student@ca647 assignment2]$
```

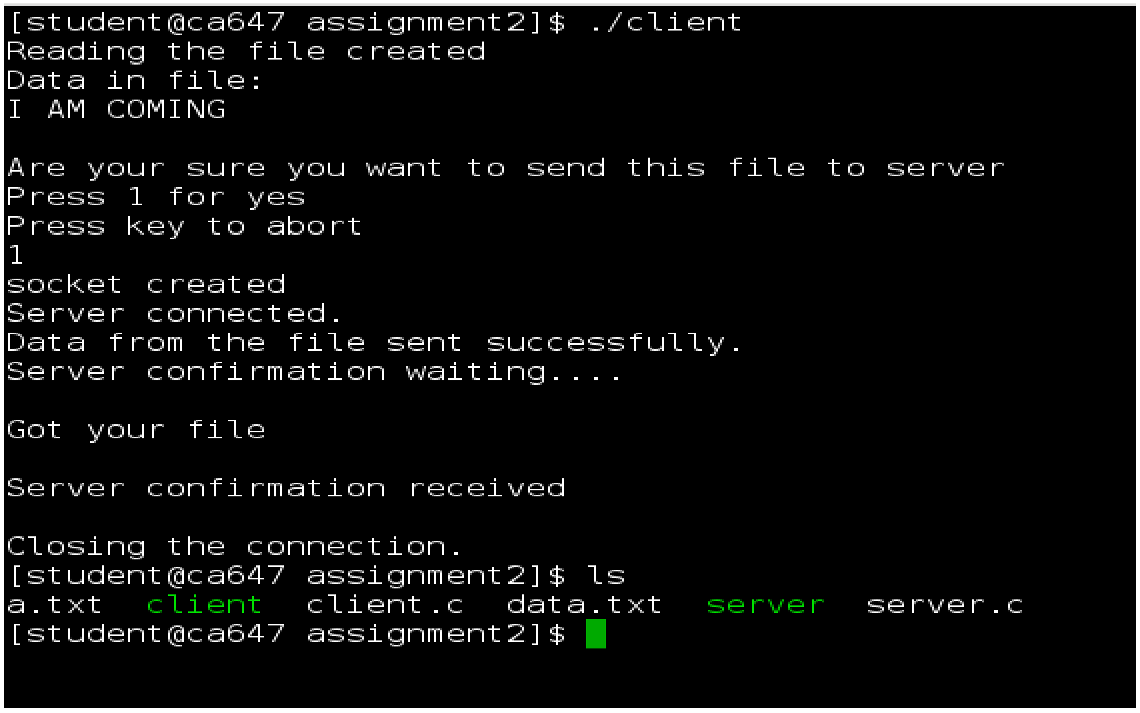
```
[student@ca647 assignment2]$ ./client
Reading the file created
Data in file:
I AM COMING

Are your sure you want to send this file to server
Press 1 for yes
Press key to abort
1
socket created
Server connected.
Data from the file sent successfully.
Server confirmation waiting....

Got your file

Server confirmation received

Closing the connection.
[student@ca647 assignment2]$ ls
a.txt  client  client.c  data.txt  server  server.c
[student@ca647 assignment2]$
```

Server



```
[student@ca647 assignment2]$ clear

[student@ca647 assignment2]$ ./server
Server socket created
Binded.
Waiting for connection...
Data written in the file successfully.
[student@ca647 assignment2]$ ls
a.txt  client  client.c  data.txt  server  server.c
[student@ca647 assignment2]$ 
```

Client



```
[student@ca647 assignment2]$ ./client
Reading the file created
Data in file:
I AM COMING

Are your sure you want to send this file to server
Press 1 for yes
Press key to abort
1
socket created
Server connected.
Data from the file sent successfully.
Server confirmation waiting....

Got your file

Server confirmation received

Closing the connection.
[student@ca647 assignment2]$ ls
a.txt  client  client.c  data.txt  server  server.c
[student@ca647 assignment2]$ 
```

NAME: VIRKANT SINGH

STUDENT NO: 21262315