B.TECH. (CSE)
IV SEMESTER

UE20CS253 – COMPUTER NETWORKS

PROJECT REPORT

ON

FTP USING TCP IN C

SUBMITTED BY:

NAME	SRN
Rohan Amin	PES2UG20CS416
Chaitra B N	PES2UG20CS424
Nidhi Torvi	PES2UG20CS445

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ELECTRONIC CITY CAMPUS,

BENGALURU – 560100, KARNATAKA, INDIA

ABSTRACT OF THE PROJECT:

An FTP server is a computer program that is built to handle data transfer between computers.

The server waits for clients to connect to it and issue commands that tell the server to upload, download, or list directories.

The FTP protocol is the commands the FTP server uses to accomplish this.

This project contains 2 main parts

1. Upload to server
2. Download from Server

The client side requests the connection from the server ..
which is acknowledge by the server by setting up TCP connection .This gives client the permission to upload the files.. and also the accesses the files that are present in the server to download them

DOWNLOAD FROM SERVER…….

CODES:

SERVER SIDE:

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>


struct sockaddr_in c_addr;
char fname[100];

void* SendFileToClient(int *arg)
{
      int connfd=(int)*arg;
      printf("Connection accepted and id: %d\n",connfd);
      printf("Connected to Clent:
%s:%d\n",inet_ntoa(c_addr.sin_addr),ntohs(c_addr.sin_port));
       write(connfd, fname,256);

        FILE *fp = fopen(fname,"rb");
        if(fp==NULL)
        {
            printf("File opern error");
            return 1;
        }

        /* Read data from file and send it */
        while(1)
        {
            /* First read file in chunks of 256 bytes */
            unsigned char buff[1024]={0};
            int nread = fread(buff,1,1024,fp);
            //printf("Bytes read %d \n", nread);

            /* If read was success, send data. */
            if(nread > 0)
            {
                //printf("Sending \n");
                write(connfd, buff, nread);
```

```
            }
            if (nread < 1024)
            {
                if (feof(fp))
        {
                    printf("End of file\n");
            printf("File transfer completed for id: %d\n",connfd);
        }
                if (ferror(fp))
                    printf("Error reading\n");
                break;
            }
        }
printf("Closing Connection for id: %d\n",connfd);
close(connfd);
shutdown(connfd,SHUT_WR);
sleep(2);
}

int main(int argc, char *argv[])
{
    int connfd = 0,err;
    pthread_t tid;
    struct sockaddr_in serv_addr;
    int listenfd = 0,ret;
    char sendBuff[1025];
    int numrv;
    size_t clen=0;

    listenfd = socket(AF_INET, SOCK_STREAM, 0);
    if(listenfd<0)
    {
      printf("Error in socket creation\n");
      exit(2);
    }

    printf("Socket retrieve success\n");

    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = htonl(INADDR_ANY);
    serv_addr.sin_port = htons(5000);

    ret=bind(listenfd, (struct sockaddr*)&serv_addr,sizeof(serv_addr));
    if(ret<0)
    {
      printf("Error in bind\n");
      exit(2);
    }
```

```
    if(listen(listenfd, 10) == -1)
    {
        printf("Failed to listen\n");
        return -1;
    }

if (argc < 2)
{
    printf("Enter file name to send: ");
        gets(fname);
}
else
   strcpy(fname,argv[1]);

    while(1)
    {
        clen=sizeof(c_addr);
        printf("Waiting...\n");
        connfd = accept(listenfd, (struct sockaddr*)&c_addr,&clen);
        if(connfd<0)
        {
      printf("Error in accept\n");
      continue;
    }
        err = pthread_create(&tid, NULL, &SendFileToClient, &connfd);
        if (err != 0)
            printf("\ncan't create thread :[%s]", strerror(err));
   }
    close(connfd);
    return 0;
}
```

CLIENT SIDE:

```
//client:

#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
```

```c
#include <unistd.h>
#include <errno.h>
#include <arpa/inet.h>
void gotoxy(int x,int y)
 {
    printf("%c[%d;%df",0x1B,y,x);
 }
int main(int argc, char *argv[])
{
    system("clear");
    int sockfd = 0;
    int bytesReceived = 0;
    char recvBuff[1024];
    memset(recvBuff, '0', sizeof(recvBuff));
    struct sockaddr_in serv_addr;

    /* Create a socket first */
    if((sockfd = socket(AF_INET, SOCK_STREAM, 0))< 0)
    {
        printf("\n Error : Could not create socket \n");
        return 1;
    }

    /* Initialize sockaddr_in data structure */
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(5000); // port
    char ip[50];
if (argc < 2)
{
    printf("Enter IP address to connect: ");
    gets(ip);
}
else
  strcpy(ip,argv[1]);

    serv_addr.sin_addr.s_addr = inet_addr(ip);

    /* Attempt a connection */
    if(connect(sockfd, (struct sockaddr *)&serv_addr, sizeof(serv_addr))<0)
    {
        printf("\n Error : Connect Failed \n");
        return 1;
    }

    printf("Connected to ip: %s :
%d\n",inet_ntoa(serv_addr.sin_addr),ntohs(serv_addr.sin_port));

     /* Create file where data will be stored */
```

```
        FILE *fp;
    char fname[100];
    read(sockfd, fname, 256);
    //strcat(fname,"AK");
    printf("File Name: %s\n",fname);
    printf("Receiving file...");
     fp = fopen(fname, "ab");
        if(NULL == fp)
        {
         printf("Error opening file");
         return 1;
        }
    long double sz=1;
    /* Receive data in chunks of 256 bytes */
    while((bytesReceived = read(sockfd, recvBuff, 1024)) > 0)
    {
        sz++;
        gotoxy(0,4);
        printf("Received: %llf Mb",(sz/1024));
    fflush(stdout);
        // recvBuff[n] = 0;
        fwrite(recvBuff, 1,bytesReceived,fp);
        // printf("%s \n", recvBuff);
    }

    if(bytesReceived < 0)
    {
        printf("\n Read Error \n");
    }
    printf("\nFile OK....Completed\n");
    return 0;
}
```

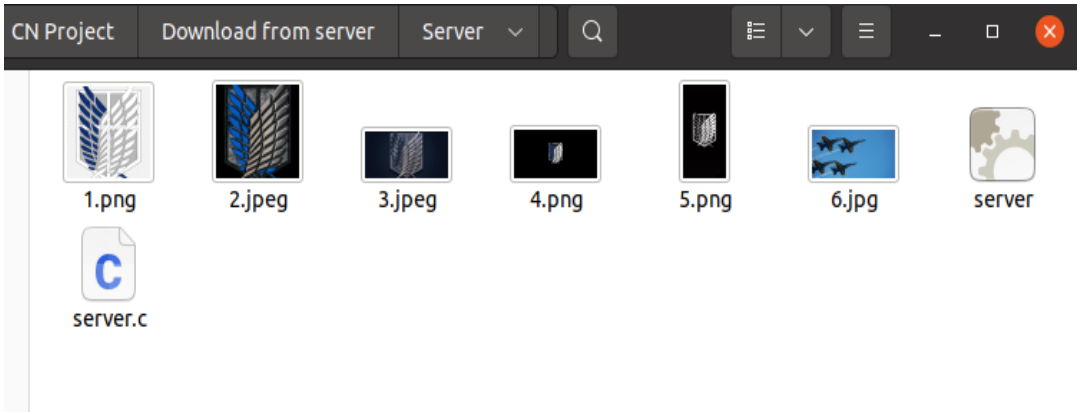
SCREEN SHOTS OF THE OUTPUT:

Server :

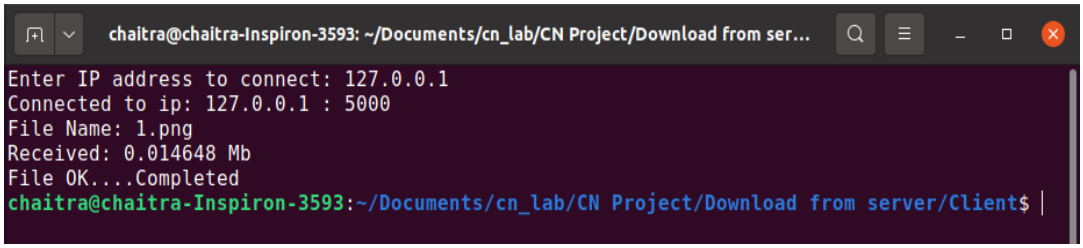
Connection is established between server and client for a certain ip address and port.

```
chaitra@chaitra-Inspiron-3593:~/Documents/cn_lab/CN Project/Download from server/Server$ ./
server
Socket retrieve success
Enter file name to send: 1.png
Waiting...
Waiting...
Connection accepted and id: 4
Connected to Clent: 127.0.0.1:56012
End of file
File transfer completed for id: 4
Closing Connection for id: 4
```

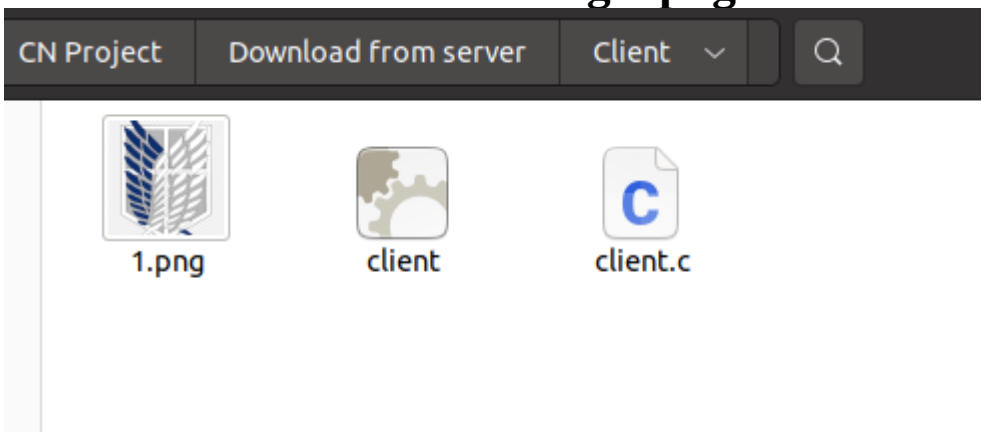
Server files:



Client:



Client files after downloading 1.png :



UPLOAD TO SERVER…..

CODES:

SERVER SIDE:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <arpa/inet.h>
#define SIZE 1024

void write_file(int sockfd){
  int n;
  FILE *fp;
  char *filename = "recv.txt";
  char buffer[SIZE];

  fp = fopen(filename, "w");
  while (1) {
    n = recv(sockfd, buffer, SIZE, 0);
    if (n <= 0){
      break;
      return;
    }
    fprintf(fp, "%s", buffer);
    bzero(buffer, SIZE);
  }
  return;
}

int main(){
  char *ip = "10.0.4.15";
  int port = 8080;
  int e;

  int sockfd, new_sock;
  struct sockaddr_in server_addr, new_addr;
  socklen_t addr_size;
  char buffer[SIZE];

  sockfd = socket(AF_INET, SOCK_STREAM, 0);
  if(sockfd < 0) {
    perror("[-]Error in socket");
    exit(1);
  }
  printf("[+]Server socket created successfully.\n");
```

```
  server_addr.sin_family = AF_INET;
  server_addr.sin_port = port;
  server_addr.sin_addr.s_addr = inet_addr(ip);

  e = bind(sockfd, (struct sockaddr*)&server_addr, sizeof(server_addr));
  if(e < 0) {
    perror("[-]Error in bind");
    exit(1);
  }
  printf("[+]Binding successfull.\n");

  if(listen(sockfd, 10) == 0){
    printf("[+]Listening....\n");
  }else{
    perror("[-]Error in listening");
    exit(1);
  }

  addr_size = sizeof(new_addr);
  new_sock = accept(sockfd, (struct sockaddr*)&new_addr, &addr_size);
  write_file(new_sock);
  printf("[+]Data written in the file successfully.\n");

  return 0;
}
```

CLIENT SIDE:

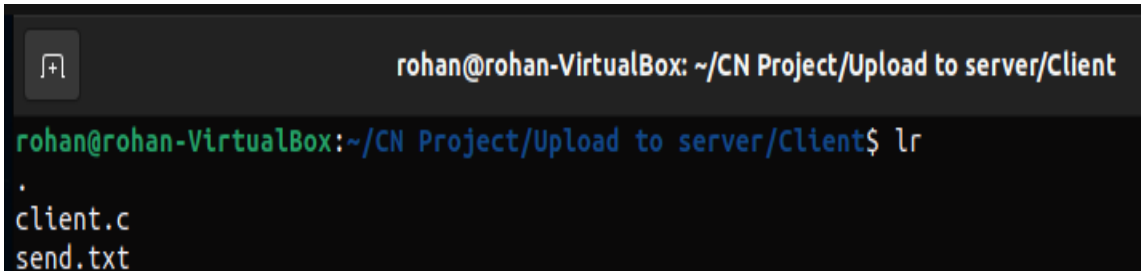
```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <arpa/inet.h>
#define SIZE 1024

void send_file(FILE *fp, int sockfd){
  int n;
  char data[SIZE] = {0};

  while(fgets(data, SIZE, fp) != NULL) {
    if (send(sockfd, data, sizeof(data), 0) == -1) {
      perror("[-]Error in sending file.");
      exit(1);
    }
    bzero(data, SIZE);
  }
}
```

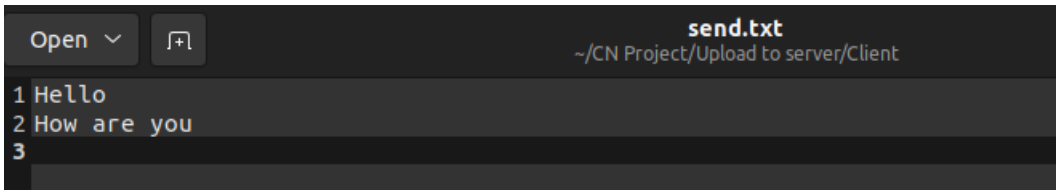
```c
int main(){
  char *ip = "10.0.4.15";
  int port = 8080;
  int e;

  int sockfd;
  struct sockaddr_in server_addr;
  FILE *fp;
  char *filename = "send.txt";

  sockfd = socket(AF_INET, SOCK_STREAM, 0);
  if(sockfd < 0) {
    perror("[-]Error in socket");
    exit(1);
  }
  printf("[+]Server socket created successfully.\n");

  server_addr.sin_family = AF_INET;
  server_addr.sin_port = port;
  server_addr.sin_addr.s_addr = inet_addr(ip);

  e = connect(sockfd, (struct sockaddr*)&server_addr, sizeof(server_addr));
  if(e == -1) {
    perror("[-]Error in socket");
    exit(1);
  }
  printf("[+]Connected to Server.\n");

  fp = fopen(filename, "r");
  if (fp == NULL) {
    perror("[-]Error in reading file.");
    exit(1);
  }

  send_file(fp, sockfd);
  printf("[+]File data sent successfully.\n");

  printf("[+]Closing the connection.\n");
  close(sockfd);

  return 0;
}
```

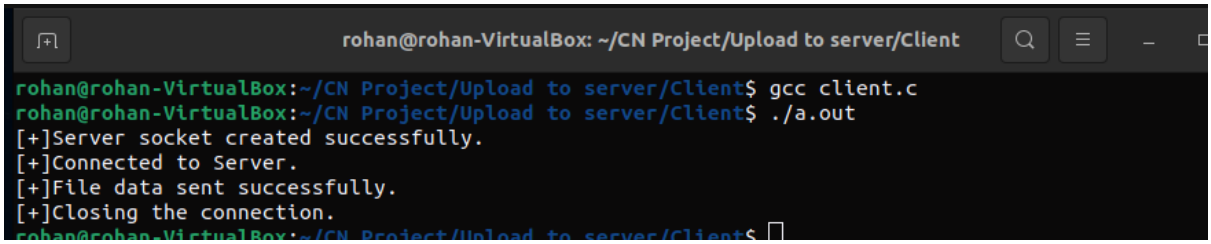
SCREEN SHOTS OF THE OUTPUT:

Client :

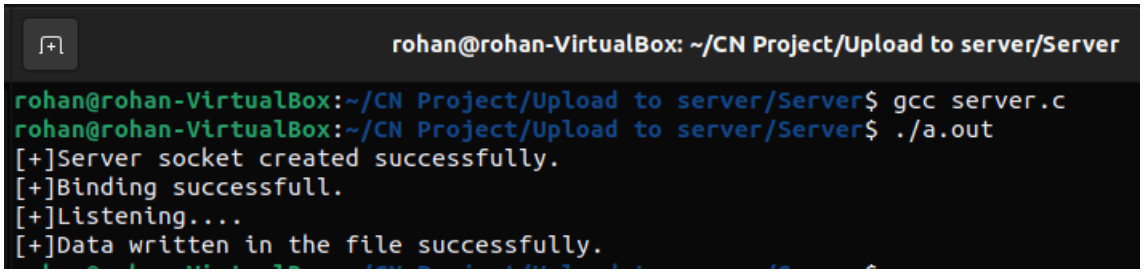
We are sending send.txt from the client to the server.



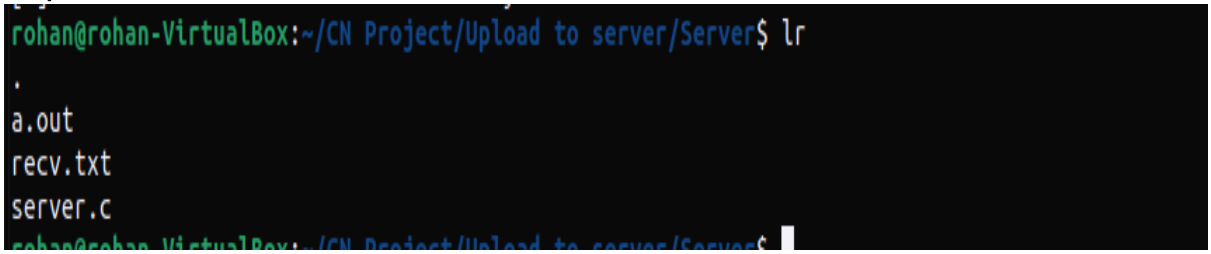




Server:



recv.txt is uploaded at the server side and the contents of send.txt is copied to it.



The recv.txt file which is uploaded to server: