3a. Design a C program to implement client server model

(TCP) using socket programming.

SERVER

#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 = "file2.txt";

char buffer[SIZE];

fp = fopen(filename, "w");

if(fp==NULL)

{

     perror("[-]Error in creating file.");

     exit(1);

}

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 = "127.0.0.1";

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. \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 Binding");

      exit(1);

  }

  printf("[+]Binding Successfull.\n");

  e = listen(sockfd, 10);

  if(e==0)

  {

      printf("[+]Listening...\n");

  }

  else

  {

      perror("[-]Error in Binding");

      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 text file ");

}

CLIENT

#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)

{

char data[SIZE] = {0};

while(fgets(data, SIZE, fp)!=NULL)

{

     if(send(sockfd, data, sizeof(data), 0)== -1)

     {

         perror("[-] Error in sendung data");

         exit(1);

     }

     bzero(data, SIZE);

}

}

int main()

{

char \*ip = "127.0.0.1";

int port = 8080;

int e;

int sockfd;

struct sockaddr\_in server\_addr;

FILE \*fp;

char \*filename = "file.txt";

  sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

if(sockfd<0)

{

     perror("[-]Error in socket");

     exit(1);

}

  printf("[+]Server socket created. \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 Connecting");

      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 send successfully. \n");

  close(sockfd);

  printf("[+]Disconnected from the server. \n");

  return 0;

}

OUTPUT:

SERVER SIDE

Socket successfully created..

Socket successfully binded..

Server listening..

server accept the client...

From client: hi

     To client : hello

From client: exit

     To client : exit

Server Exit...

CLIENT SIDE

Socket successfully created..

connected to the server..

Enter the string : hi

From Server : hello

Enter the string : exit

From Server : exit

Client Exit...