COMPUTER NETWORKS AND SECURITY LABORATORY

Group B

Assignment No. 9

NAME :- OJUS P. JAISWAL

ROLL NO. :- TACO19108

YEAR AND DIV :- TE A

Ques :- Write a program using TCP socket for wired network for following

a. Say Hello to Each other

b. File transfer

c. Calculator

Solution :-

Program :

a) Say Hello to Each other =>

1) Server

```
// Hello Server-Side Program

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

#define buffsize  150

int main(void)
{
  struct sockaddr_in servaddr,clientaddr;
  char buff[buffsize],crcbuff[buffsize];
  int listenfd,connfd;
  int sin_size;
```

```
 if((listenfd=socket(AF_INET,SOCK_STREAM,0))==-1)
            perror("Socket Creation Error.\n");
 else
            printf("Socket Created Successfully\n");
 bzero((char *) &servaddr, sizeof(servaddr));
 servaddr.sin_family=AF_INET;
 servaddr.sin_addr.s_addr=inet_addr("127.0.0.1");
 servaddr.sin_port=htons(5000);
 if(bind(listenfd, (struct sockaddr *)&servaddr,
                                                          sizeof(servaddr)) == -1)
           perror("Bind Error\n");
 listen(listenfd,4);
 sin_size = sizeof(struct sockaddr_in);
 for(;;)
 {
 if((connfd=accept(listenfd,(struct sockaddr *)&clientaddr,
                                    &sin_size))==-1)
       perror("Accept Error\n");
           strcpy(buff,"Hello Client - Server\n");
           write(connfd,buff,strlen(buff));
           close(connfd);
 }
}
```

2) Client

```
// Hello Client Side

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

#define buffsize  150

int main(void)
{
 struct sockaddr_in clientaddr;
 char recvline[buffsize];
```

```
 int sockfd,n;
 if((sockfd=socket(AF_INET,SOCK_STREAM,0))<0)
          perror("Socket Creation Error.\n");
 else
          printf("Socket Created Successfully\n");
 bzero((char *) &clientaddr, sizeof(clientaddr));
 clientaddr.sin_family=AF_INET;
 clientaddr.sin_addr.s_addr=inet_addr("127.0.0.1");
 clientaddr.sin_port=htons(5000);
 if(connect(sockfd,(struct sockaddr *) &clientaddr, sizeof(clientaddr)) < 0 )
          perror("Connect error\n");
 else
         printf("Connected successfully\n");
 while(( n = read(sockfd,recvline,buffsize)) > 0 )
 {
         recvline[n]=0;
         if(fputs(recvline,stdout)==EOF)
          perror("fputs Error.\n");
 }
 if(n < 0)
  perror("Read Error\n");
 exit(0);
}
```

b) File transfer =>

1) Server

```
import java.io.BufferedInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.OutputStream;
import java.net.InetAddress;
import java.net.ServerSocket;
import java.net.Socket;

public class TcpFSer {

   public static void main(String[] args) throws Exception {
      //Initialize Sockets
      ServerSocket ssock = new ServerSocket(5000);
      Socket socket = ssock.accept();

      //The InetAddress specification
      InetAddress IA = InetAddress.getByName("localhost");

      //Specify the file
      File file = new File("/home/dypiemr-/Desktop/Ashish_Shahane/Ashish.txt");
      FileInputStream fis = new FileInputStream(file);
      BufferedInputStream bis = new BufferedInputStream(fis);

      //Get socket's output stream
      OutputStream os = socket.getOutputStream();

      //Read File Contents into contents array
      byte[] contents;
      long fileLength = file.length();
      long current = 0;

      long start = System.nanoTime();
      while(current!=fileLength){
         int size = 10000;
         if(fileLength - current >= size)
            current += size;
         else{
```

```
            size = (int)(fileLength - current);
            current = fileLength;
          }
          contents = new byte[size];
          bis.read(contents, 0, size);
          os.write(contents);
          System.out.print("Sending file ... "+(current*100)/fileLength+"% complete!");
        }

        os.flush();
        //File transfer done. Close the socket connection!
        socket.close();
        ssock.close();
        System.out.println("File sent succesfully!");
    }
  }
```

2) Client

```
import java.io.BufferedOutputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.net.InetAddress;
import java.net.Socket;


public class TcpFCli {

   public static void main(String[] args) throws Exception{

      //Initialize socket
      Socket socket = new Socket(InetAddress.getByName("localhost"), 5000);
      byte[] contents = new byte[10000];

      //Initialize the FileOutputStream to the output file's full path.
      FileOutputStream fos = new FileOutputStream("/home/dypiemr-
/Desktop/Ashish_Shahane/AshishRecived.txt");
      BufferedOutputStream bos = new BufferedOutputStream(fos);
      InputStream is = socket.getInputStream();
```

```
        //No of bytes read in one read() call
        int bytesRead = 0;

        while((bytesRead=is.read(contents))!=-1)
            bos.write(contents, 0, bytesRead);

        bos.flush();
        socket.close();

        System.out.println("File saved successfully!");
    }
}
```

c) Calculator =>

1) Server

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

void main()
{
int b,sockfd,connfd,sin_size,l,n,len;
char operator;
int op1,op2,result;
if((sockfd=socket(AF_INET,SOCK_STREAM,0))>0)
printf("socket created sucessfully\n");  //socket creation
//printf("%d\n", sockfd);               //on success 0 otherwise -1

struct sockaddr_in servaddr;
struct sockaddr_in clientaddr;

servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=inet_addr("127.0.0.1");
servaddr.sin_port=6006;

if((bind(sockfd, (struct sockaddr *)&servaddr,sizeof(servaddr)))==0)
printf("bind sucessful\n");   //bind() assigns the
    // address specified by addr to the socket referred to by the file
     // descriptor sockfd. addrlen specifies the size, in bytes, of the
    // address structure pointed to by addr.  Traditionally, this operation is
     // called “assigning a name to a socket”.

//printf("%d\n",b);

if((listen(sockfd,5))==0) //listen for connections on a socket
printf("listen sucessful\n");
```

```
//printf("%d\n",l);

sin_size = sizeof(struct sockaddr_in);
if((connfd=accept(sockfd,(struct sockaddr *)&clientaddr,&sin_size))>0);
printf("accept sucessful\n");
//printf("%d\n",connfd);
read(connfd, &operator,10);
read(connfd,&op1,sizeof(op1));
read(connfd,&op2,sizeof(op2));
switch(operator) {
      case '+': result=op1 + op2;
       printf("Result is: %d + %d = %d\n",op1, op2, result);
       break;
      case '-':result=op1 - op2;
            printf("Result is: %d - %d = %d\n",op1, op2, result);
            break;
      case '*':result=op1 * op2;
            printf("Result is: %d * %d = %d\n",op1, op2, result);
            break;
      case '/':result=op1 / op2;
            printf("Result is: %d / %d = %d\n",op1, op2, result);
            break;
      default:
            printf("ERROR: Unsupported Operation");
  }

write(connfd,&result,sizeof(result));
close(sockfd);
}
```

2) Client

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

//#define buffsize  150
void main()
```

```
{
int b,sockfd,sin_size,con,n,len;
//char buff[256];
char operator;
int op1,op2,result;
if((sockfd=socket(AF_INET,SOCK_STREAM,0))>0)
printf("socket created sucessfully\n");
//printf("%d\n", sockfd);
struct sockaddr_in servaddr;

servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=inet_addr("127.0.0.1");
servaddr.sin_port=6006;

sin_size = sizeof(struct sockaddr_in);
if((con=connect(sockfd,(struct sockaddr *) &servaddr, sin_size))==0); //initiate a connection on
a socket
printf("connect sucessful\n");
printf("Enter operation:\n +:Addition \n -: Subtraction \n /: Division \n*:Multiplication \n");
scanf("%c",&operator);
printf("Enter operands:\n");
scanf("%d %d", &op1, &op2);

write(sockfd,&operator,10);
write(sockfd,&op1,sizeof(op1));
write(sockfd,&op2,sizeof(op2));
read(sockfd,&result,sizeof(result));
printf("Operation result from server=%d\n",result);
close(sockfd);
}
```

Output :

a) Say Hello to Each other =>

1) Server

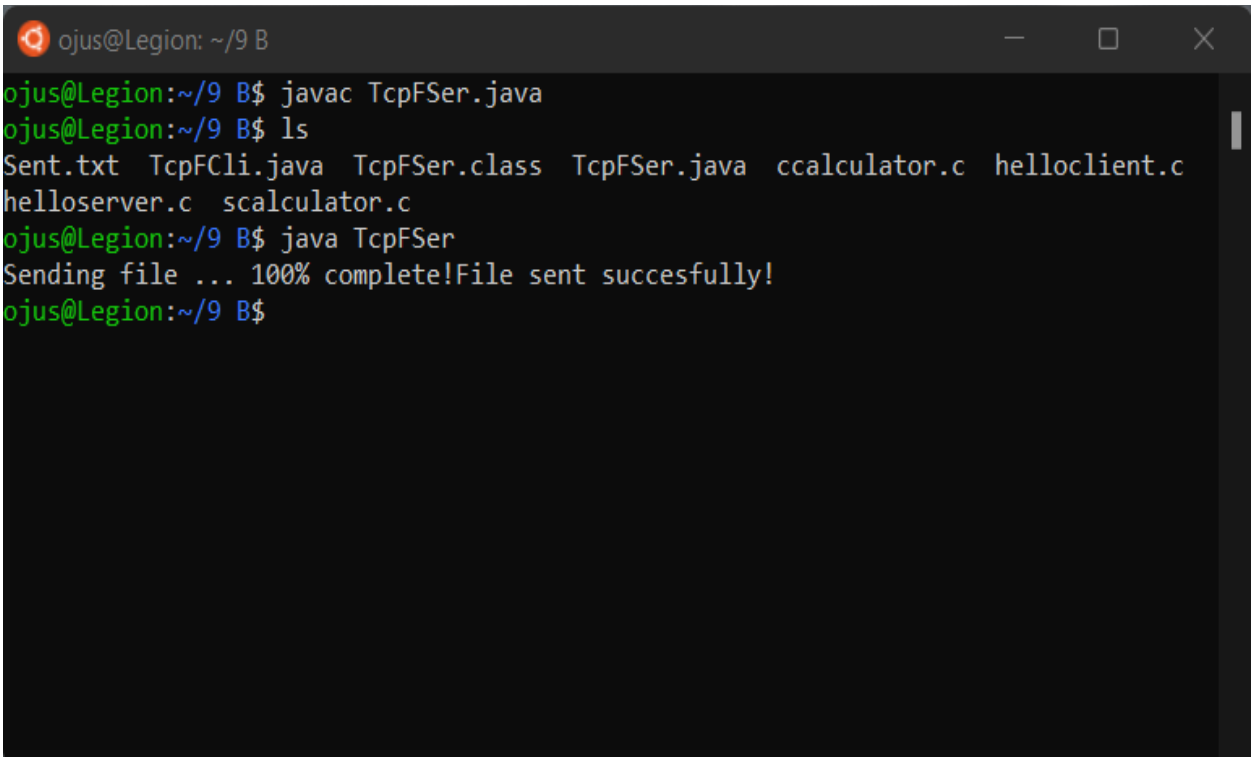
```
ojus@Legion: ~/9 B
ojus@Legion:~/9 B$ gcc -o server helloserver.c
helloserver.c: In function ‘main’:
helloserver.c:27:28: warning: implicit declaration of function ‘inet_addr’
[-Wimplicit-function-declaration]
   27 |    servaddr.sin_addr.s_addr=inet_addr("127.0.0.1");
      |                             ^~~~~~~~~~
helloserver.c:42:3: warning: implicit declaration of function ‘write’; did
you mean ‘fwrite’? [-Wimplicit-function-declaration]
   42 |   write(connfd,buff,strlen(buff));
      |   ^~~~~
      |   fwrite
helloserver.c:43:3: warning: implicit declaration of function ‘close’; did
you mean ‘pclose’? [-Wimplicit-function-declaration]
   43 |   close(connfd);
      |   ^~~~~
      |   pclose
ojus@Legion:~/9 B$ ./server
Socket Created Successfully
```

2) Client

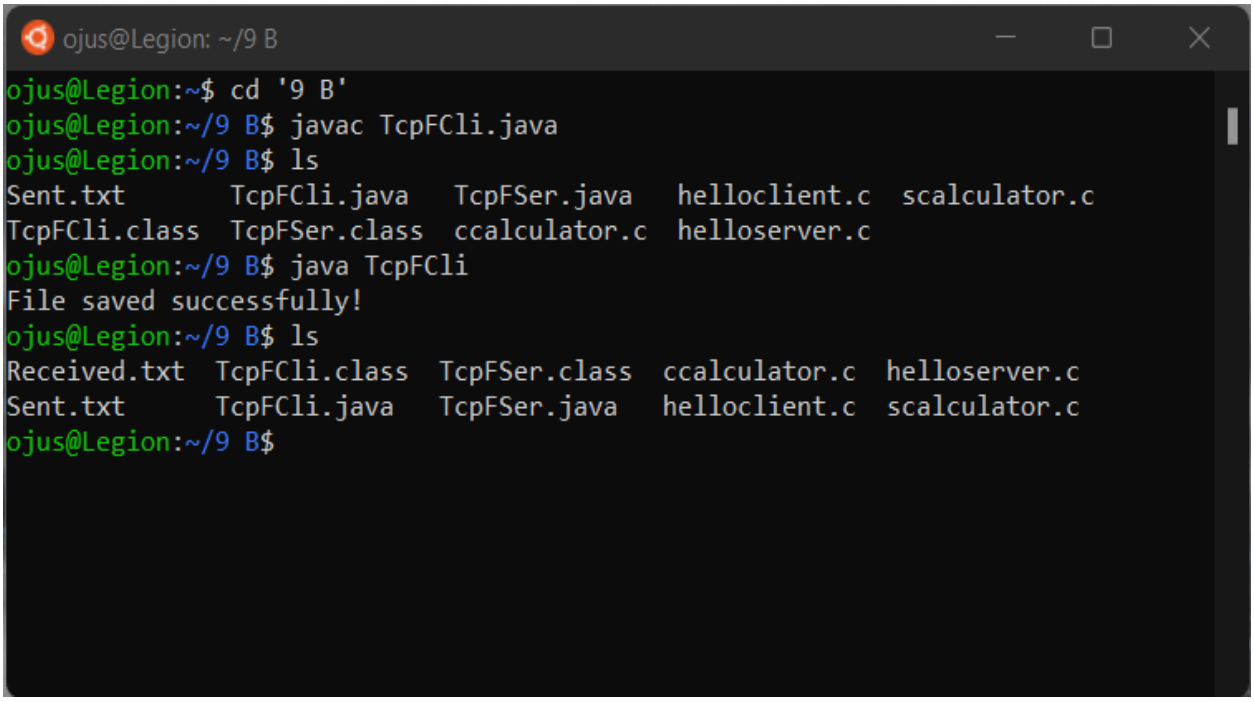
```
ojus@Legion: ~/9 B
ojus@Legion:~$ cd '9 B'
ojus@Legion:~/9 B$ gcc -o client helloclient.c
helloclient.c: In function ‘main’:
helloclient.c:30:15: warning: implicit declaration of function ‘read’; did
you mean ‘fread’? [-Wimplicit-function-declaration]
   30 |   while(( n = read(sockfd,recvline,buffsize)) > 0 )
      |               ^~~~
      |               fread
ojus@Legion:~/9 B$ ./client
Socket Created Successfully
Connected successfully
Hello Client - Server
ojus@Legion:~/9 B$
```

b) File transfer =>

1) Server

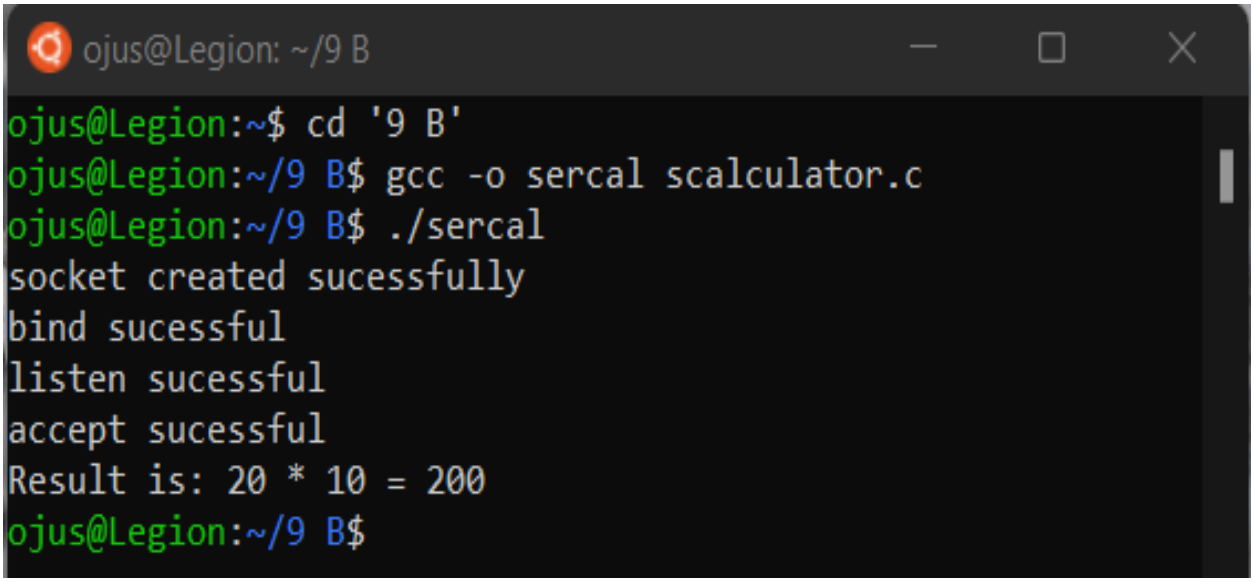


2) Client



c) Calculator =>

1) Server



2) Client