**CSE1004 LAB – L52+L53 – Dr Kanchana Devi V**

**Lab Submission – 6**

**[ADDITIONAL EXERCISE]**

**Done by:** ARVIND CB 19BCE1221

**Server Code:**

#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>

#include <time.h>

int main(int argc, char \*argv[])

{

int listenfd = 0, connfd = 0;

struct sockaddr\_in serv\_addr;

char sendBuff[1025];

time\_t ticks;

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

memset(&serv\_addr, '0', sizeof(serv\_addr));

memset(sendBuff, '0', sizeof(sendBuff));

serv\_addr.sin\_family = AF\_INET;

serv\_addr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

serv\_addr.sin\_port = htons(5000);

bind(listenfd, (struct sockaddr\*)&serv\_addr, sizeof(serv\_addr));

listen(listenfd, 10);

int data[10];

int dataatrec[10],c,c1,c2,c3,i;

printf("Enter 4 bits of data one by one\n");

scanf("%d",&data[0]);

scanf("%d",&data[1]);

scanf("%d",&data[2]);

scanf("%d",&data[4]);

data[6]=data[0]^data[2]^data[4];

data[5]=data[0]^data[1]^data[4];

data[3]=data[0]^data[1]^data[2];

printf("\nEncoded data is\n");

for(i=0;i<7;i++)

printf("%d",data[i]);

printf("\n\nEnter received data bits one by one\n");

for(i=0;i<7;i++)

scanf("%d",&dataatrec[i]);

c1=dataatrec[6]^dataatrec[4]^dataatrec[2]^dataatrec[0];

c2=dataatrec[5]^dataatrec[4]^dataatrec[1]^dataatrec[0];

c3=dataatrec[3]^dataatrec[2]^dataatrec[1]^dataatrec[0];

c=c3\*4+c2\*2+c1 ;

if(c==0) {

printf("\nNo error while transmission of data\n");

}

else {

printf("\nError on position %d",c);

printf("\nData sent : ");

for(i=0;i<7;i++)

printf("%d",data[i]);

printf("\nData received : ");

for(i=0;i<7;i++)

printf("%d",dataatrec[i]);

printf("\nCorrect message is\n");

if(dataatrec[7-c]==0)

dataatrec[7-c]=1;

else

dataatrec[7-c]=0;

for (i=0;i<7;i++) {

printf("%d",dataatrec[i]);

}

}

}

**Client code:**

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <netdb.h>

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <unistd.h>

#include <errno.h>

#include <arpa/inet.h>

int main(int argc, char \*argv[])

{

int sockfd = 0, n = 0;

char recvBuff[1024];

struct sockaddr\_in serv\_addr;

if(argc != 2)

{

printf("\n Usage: %s <ip of server> \n",argv[0]);

return 1;

}

memset(recvBuff, '0',sizeof(recvBuff));

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

{

printf("\n Error : Could not create socket \n");

return 1;

}

memset(&serv\_addr, '0', sizeof(serv\_addr));

serv\_addr.sin\_family = AF\_INET;

serv\_addr.sin\_port = htons(5000);

if(inet\_pton(AF\_INET, argv[1], &serv\_addr.sin\_addr)<=0)

{

printf("\n inet\_pton error occured\n");

return 1;

}

if( connect(sockfd, (struct sockaddr \*)&serv\_addr, sizeof(serv\_addr)) < 0)

{

printf("\n Error : Connect Failed \n");

return 1;

}

while ( (n = read(sockfd, recvBuff, sizeof(recvBuff)-1)) > 0)

{

recvBuff[n] = 0;

if(fputs(recvBuff, stdout) == EOF)

{

printf("\n Error : Fputs error\n");

}

}

if(n < 0)

{

printf("\n Read error \n");

}

int dataatrec[10],c,c1,c2,c3,i;

int test[10];

int p , p1 , p2 , p3 ,p4;

printf("data recieved is:");

for(int i=0;i<7;i++)

{

n = read(listenfd , &dataatrec[i], sizeof(int));

if(n<0)

perror("error on recieving data");

printf("%d" , dataatrec[i]);

}

printf("\n\nEnter received data bits one by one\n");

for(i=0;i<7;i++)

scanf("%d",&dataatrec[i]);

c1=dataatrec[6]^dataatrec[4]^dataatrec[2]^dataatrec[0];

c2=dataatrec[5]^dataatrec[4]^dataatrec[1]^dataatrec[0];

c3=dataatrec[3]^dataatrec[2]^dataatrec[1]^dataatrec[0];

c=c3\*4+c2\*2+c1 ;

if(c==0) {

printf("\nNo error while transmission of data\n");

}

else {

printf("\nError on position %d",c);

printf("\nData sent : ");

for(i=0;i<7;i++)

printf("%d",dataatrec[i]);

printf("\nData received : ");

for(i=0;i<7;i++)

printf("%d",dataatrec[i]);

printf("\nCorrect message is\n");

if(dataatrec[7-c]==0)

dataatrec[7-c]=1;

else

dataatrec[7-c]=0;

for (i=0;i<7;i++) {

printf("%d",dataatrec[i]);

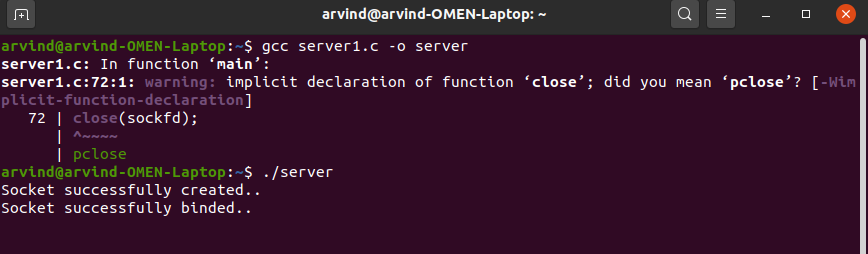
}

}

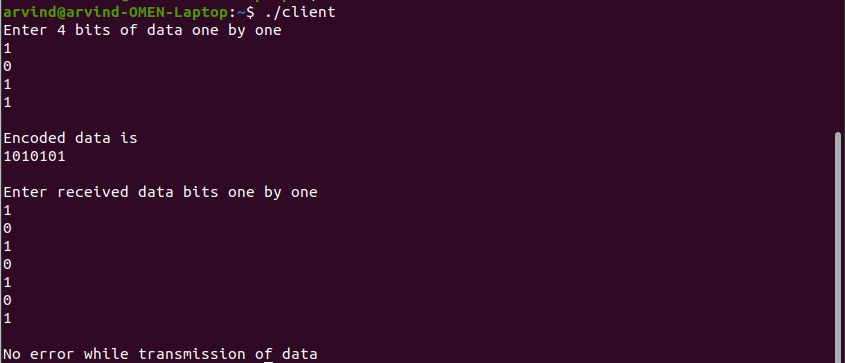
return 0;

}

**OUTPUT SCREENSHOTS**

 **If there is an error**

 **If transmission is successful**

****