**QN1: a.Sender**

//Suvash Chandra Yadav

#include "stdio.h"

#include "sys/types.h"

#include "netinet/in.h"

#include "string.h"

#include "sys/socket.h"

#include "stdlib.h"

#include "unistd.h"

int main() {

int PORT = 8880;

int sd,i,r,bi,nsd;

int frame,prev\_frame=0,count=0;;

char ack[5],buff[30];

struct sockaddr\_in ser,cli;

if((sd=socket(AF\_INET,SOCK\_STREAM,IPPROTO\_TCP))==-1)

{

printf("\n Sorry! The Socket is not created");

return 0;

}

printf("\n You may proceed, The Socket is created\n");

bzero((char\*)&cli,sizeof(ser));

printf("\n Port Address is %d\n:",PORT);

ser.sin\_family=AF\_INET;

ser.sin\_port=htons(PORT);

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

bi=bind(sd,(struct sockaddr \*)&ser,sizeof(ser));

if(bi==-1) {

printf("\n Binding ERRORRRRR!!!!!");

return 0;

}

i=sizeof(cli);

listen(sd,5);

nsd = accept(sd,((struct sockaddr \*)&cli),&i);

if(nsd==-1)

{

printf("\nKindly, check the description parameters\n");

return 0;

}

printf("\nConnection has been established");

while(count<5)

{

ph:

printf("\n Frame %d was sent to the Receiver...\n",prev\_frame); snprintf(buff,sizeof(buff),"%d",prev\_frame);

send(nsd,buff,30,0);

r = recv(nsd,ack,5,0);

if(strcmp(ack,"ack")==0 || strcmp(ack,"ACK")==0)

{

count++;

if(prev\_frame==0) prev\_frame=1;

else prev\_frame = 0;

}

else if(strcmp(ack,"nak")==0 || strcmp(ack,"NAK")==0) {

printf("\n NAK: So again sending the Previous frame...\n"); goto ph;

}

}

printf("\n This Ends the conversation !");

send(nsd,"EOF",4,0);

close(sd);

close(nsd);

return 0;

}

**Q.N1a. Receiver**

// Suvash Chandra Yadav, 119CS0178

#include "stdio.h"

#include "sys/types.h"

#include "string.h"

#include "sys/socket.h"

#include "netinet/in.h"

#include "stdlib.h"

#include "unistd.h"

int main() {

int PORT = 8880;

int sd,con,i;

char content[30],ack[3];

struct sockaddr\_in cli;

if((sd=socket(AF\_INET,SOCK\_STREAM,IPPROTO\_TCP))==-1)

{

printf("\n Socket problem");

return 0;

}

bzero((char\*)&cli,sizeof(cli));

cli.sin\_family = AF\_INET;

cli.sin\_port=htons(PORT);

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

con=connect(sd,(struct sockaddr\*)&cli,sizeof(cli));

if(con==-1)

{

printf("\nConnection failed!!!!");

return 0;

}

i = recv(sd,content,30,0);

while(strcmp(content,"EOF") != 0)

{

printf("Frame Received : %s \n",content);

ph:

printf("\n Acknowledgement :| ");

scanf("%s",ack);

if(!(strcmp(ack,"ack")==0 || strcmp(ack,"nak")==0 || strcmp(ack,"ACK")==0 || strcmp(ack,"NAK")==0))

{

printf("\n Invalid Acknowledgement..use ACK or NAK..\n ");

goto ph;

}

send(sd,ack,5,0);

i=recv(sd,content,30,0);

}

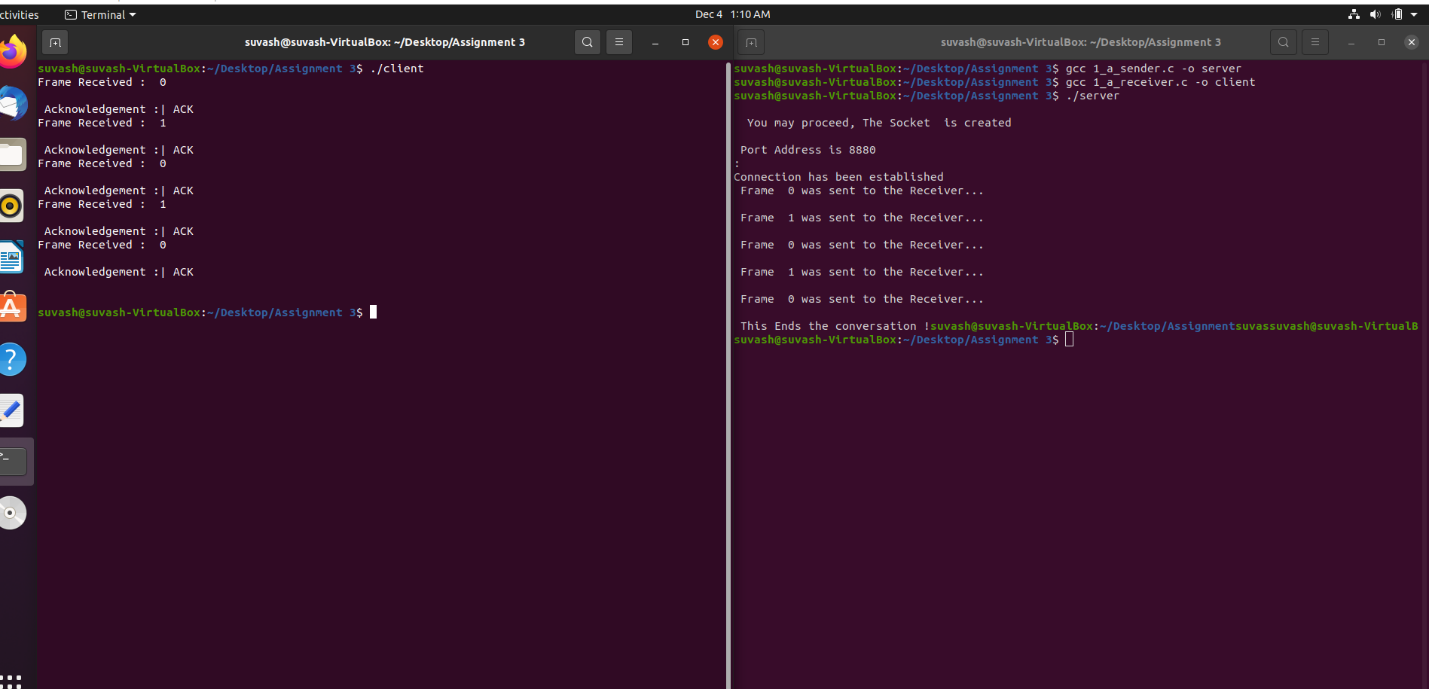
printf("\n\n This ends the conversation !");

close(sd);

return 0;

}

***OUTPUT:1a.***



***Q.N1b.***

#include<bits/stdc++.h>

#include<ctime>

#include<cstdlib>

using namespace std;

int main()

{

int numFrames, w;

srand(time(NULL));

cout<<"Enter the number of frames to be sent: ";

cin>>numFrames;

cout<<"\nEnter window size: ";

cin>>w;

cout<<endl;

int i = 1;

while(i <= numFrames)

{

int received = 0;

for(int j = i; j < i + w && j <= numFrames; j++)

{

cout<<"Sent Frame: "<<j<<endl;

}

cout<<endl;

for(int j = i;j < i + w && j <= numFrames; j++)

{

int ack = rand() % 2;

if(ack == 1)

{

cout<<"Acknowledgment received for frame "<<j<<endl;

received++;

}

else

{

cout<<"Acknowledgement for frame "<<j<<" not received"<<endl;

cout<<"\nRetransmitting Frames"<<endl;

break;

}

}

cout<<endl;

i += received;

}

return 0;

}

***Outputs1b.***