The following is linux source code for a TCP/IP example

Client Code

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <netdb.h>

#include <stdio.h>

#include <iostream>

using namespace std;

#define INVALID\_SOCKET -1

#define SOCKET\_ERROR -1

int main()

{

struct sockaddr\_in SvrAddr;

int ClientSocket;

if ((ClientSocket = socket(AF\_INET, SOCK\_STREAM, 0)) == INVALID\_SOCKET)

return 1;

struct hostent \*host;

if ((host = gethostbyname("localhost")) == NULL)

{

cout << "ERROR: Failed to resolve local host" << endl;

return(1); //terminate the program

}

SvrAddr.sin\_family = AF\_INET;

SvrAddr.sin\_port = htons(27000);

SvrAddr.sin\_addr.s\_addr = \*((unsigned long \*)host->h\_addr\_list[0]);

//Connect socket to specified server

if ((connect(ClientSocket, (struct sockaddr \*)&SvrAddr,

sizeof(SvrAddr))) == SOCKET\_ERROR)

{

close(ClientSocket);

return 1;

}

char TxBuffer[50] = { "\n" };

while (1)

{

cout << "Enter A String (x to terminate)" << endl;

cin >> TxBuffer;

send(ClientSocket, TxBuffer, sizeof(TxBuffer), 0);

if (TxBuffer[0] == 'x')

break;

}

close(ClientSocket);

}

Server Code

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <stdio.h>

#include <iostream>

#define INVALID\_SOCKET -1

#define SOCKET\_ERROR -1

int main()

{

struct sockaddr\_in SvrAddr;

int WelcomeSocket, ConnectionSocket;

//Data buffers

char RxBuffer[128] = {};

//create welcoming socket at port and bind local address

if ((WelcomeSocket = socket(AF\_INET, SOCK\_STREAM, 0)) == INVALID\_SOCKET)

return 1;

SvrAddr.sin\_family = AF\_INET;

SvrAddr.sin\_addr.s\_addr = INADDR\_ANY;

SvrAddr.sin\_port = htons(27000);

if ((bind(WelcomeSocket, (struct sockaddr \*)&SvrAddr,

sizeof(SvrAddr))) == SOCKET\_ERROR)

{

close(WelcomeSocket);

return 1;

}

//Specify the maximum number of clients that can be queued

if (listen(WelcomeSocket, 1) == SOCKET\_ERROR)

{

close(WelcomeSocket);

return 1;

}

std::cout << "Waiting for client connection\n" << std::endl;

ConnectionSocket = SOCKET\_ERROR;

while (1)

{

//wait for an incoming connection from a client

if ((ConnectionSocket = accept(WelcomeSocket, NULL, NULL)) == SOCKET\_ERROR)

{

return 1;

}

else

{

std::cout << "Connection Established" << std::endl;

while (1)

{

int n = recv(ConnectionSocket, RxBuffer, sizeof(RxBuffer), 0);

std::cout << "Msg Rx: " << RxBuffer << std::endl;

//Check to see if connection should be terminated

if (RxBuffer[0] == 'x')

break;

}

close(ConnectionSocket);

}

}

return 1;

}

These libraries are for Eclipse

//Used for TCP/IP Network Communications

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <netdb.h>