**Date:** 19/09/2016

**Roll No. and Name:** 14BCE154 Chauhan Prasiddh J.

14BCE151 Baldaniya Rushabhkumar H.

**Course Code and Name:** Open Source Development Lab (CE703)

**Practical No.:** 2 (Perl)

**AIM:** To develop a mini project in perl for Chat application using Socket Programming

**Methodology followed (Code):**

**Code 1: Server.pl**

#!/usr/bin/perl

#tcpserver.pl

use IO::Socket::INET;

# flush after every write

$| = 1;

my ($socket,$client\_socket);

my ($peeraddress,$peerport);

# creating object interface of IO::Socket::INET modules which internally does

# socket creation, binding and listening at the specified port address.

$socket = new IO::Socket::INET (

LocalHost => '127.0.0.1',

LocalPort => '5000',

Proto => 'tcp',

Listen => 5,

Reuse => 1

) or die "ERROR in Socket Creation : $!\n";

print "SERVER Waiting for client connection on port 5000";

while(1)

{

# waiting for new client connection.

$client\_socket = $socket->accept();

# get the host and port number of newly connected client.

$peer\_address = $client\_socket->peerhost();

$peer\_port = $client\_socket->peerport();

# write operation on the newly accepted client.

my $data;

# we can also send the data through IO::Socket::INET module,

# $client\_socket->send($data);

# read operation on the newly accepted client

# we can also read from socket through recv() in IO::Socket::INET

# $client\_socket->recv($data,1024);

while ($data = <$client\_socket>) {

print "\nClient : $data";

print "\nServer : ";

$temp = <>;

$client\_socket->send($temp);

}

}

$socket->close();

**Code 2: Client.pl**

#!/usr/bin/perl

#tcpclient.pl

use IO::Socket::INET;

# flush after every write

$| = 1;

my ($socket,$client\_socket);

# creating object interface of IO::Socket::INET modules which internally creates

# socket, binds and connects to the TCP server running on the specific port.

$socket = new IO::Socket::INET (

PeerHost => '127.0.0.1',

PeerPort => '5000',

Proto => 'tcp',

) or die "ERROR in Socket Creation : $!\n";

print "TCP Connection Success.\n";

# read the socket data sent by server.

# we can also read from socket through recv() in IO::Socket::INET

# $socket->recv($data,1024);

# write on the socket to server.

# we can also send the data through IO::Socket::INET module,

while(1) {

print "\nClient : ";

$temp = <>;

$socket->send($temp);

sleep (1);

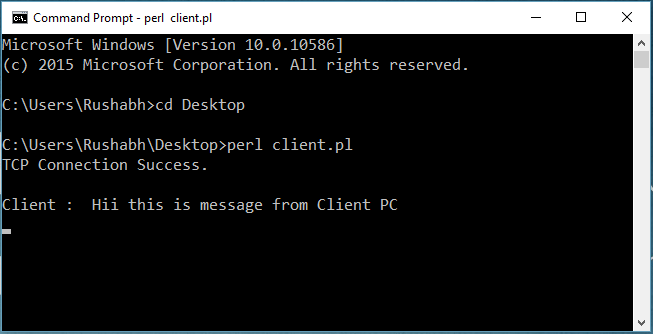
$socket->recv($data,1024);

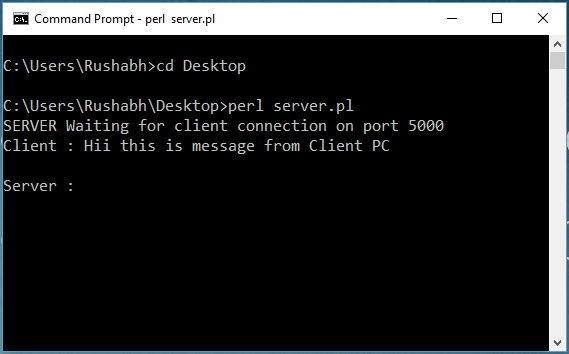
print "\nServer : $data ";

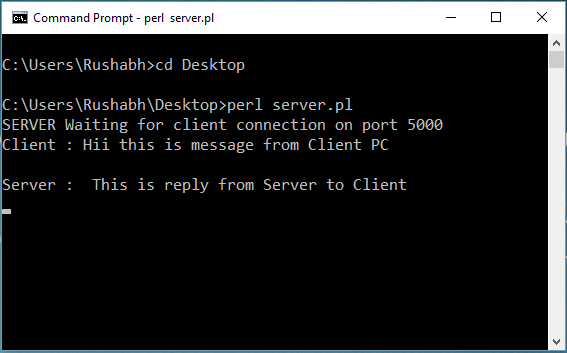
}

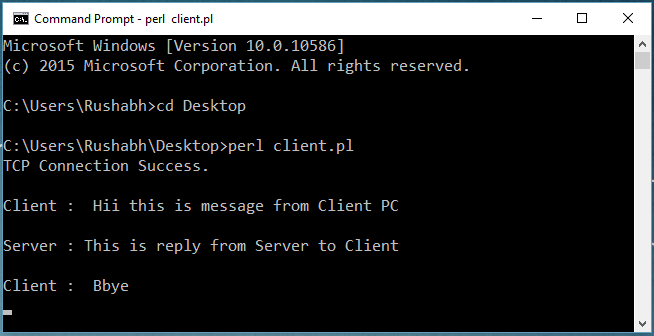
$socket->close();

**Input – output:**





****



**Conclusions:**

In this project we explored the development of Perl programming and how to implement chat application with the use of socket programming in Perl. Here the main task and the challenge is to establish the secure connection between server and the client using the Transmission Control Protocol (TCP) with local host 127.0.0.1 using the port 5000. After successfully establishing the connection with the client, both can chat with each other. So in nutshell, We learnt so many things in Perl coding through this mini project offered in this course.

**Signature of Teacher:**