|  |  |  |  |
| --- | --- | --- | --- |
| **APPROVALS** | | | |
| **Name/Department** | **Signature** | **Name/Department** | **Signature** |
| Coop Validation Engineer:  **Da Zhi** |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **HISTORY OF CHANGES** | | | |
| **Revision** | **Date** | **ECN #** | **Description** |
| 1.0 | Nov. 18, 2016 |  | Initial Release |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Linux-Windows communication module in common TCL structure**

**(Using TCL socket programming)**

# Introduction

## Background

Some of test projects which implemented in TCL script need validation and simulation at same time. However, the simulation system already exists and well-functional in Linux operation system, which means we need rewrite simulation system into windows type if we want to run both validation and simulation under Windows OS. To solve these problems, author designed a communication mechanism based on TCL socket programming between Window and Linux system, so that the validation projects in Windows system can automatically call simulation projects under Linux system.

## Architecture

The whole communication structure including 4 TCL files, “client.tcl”, “server.tcl”, “user.tcl” and “support.tcl”. In Windows system, client port will be established by running program “client.tcl” or “user.tcl”, the difference of these two programs will be explained in section 2.2, to send request message to Linux server. In Linux system, server port will be established by “server.tcl” which takes charge of receiving connection request and data from client port. “support.tcl” stored some functions will be called by other files.

## Implementation

In Linux server:

1. Source “server.tcl”

In Windows client:

1. Source “user.tcl”
2. Input the client name in “cmd”
3. Send message to server in “cmd”

# Communication structure

## Description

The following schematic diagram shows structure and whole procedure of Linux and Windows machine control in the TCL test architecture.

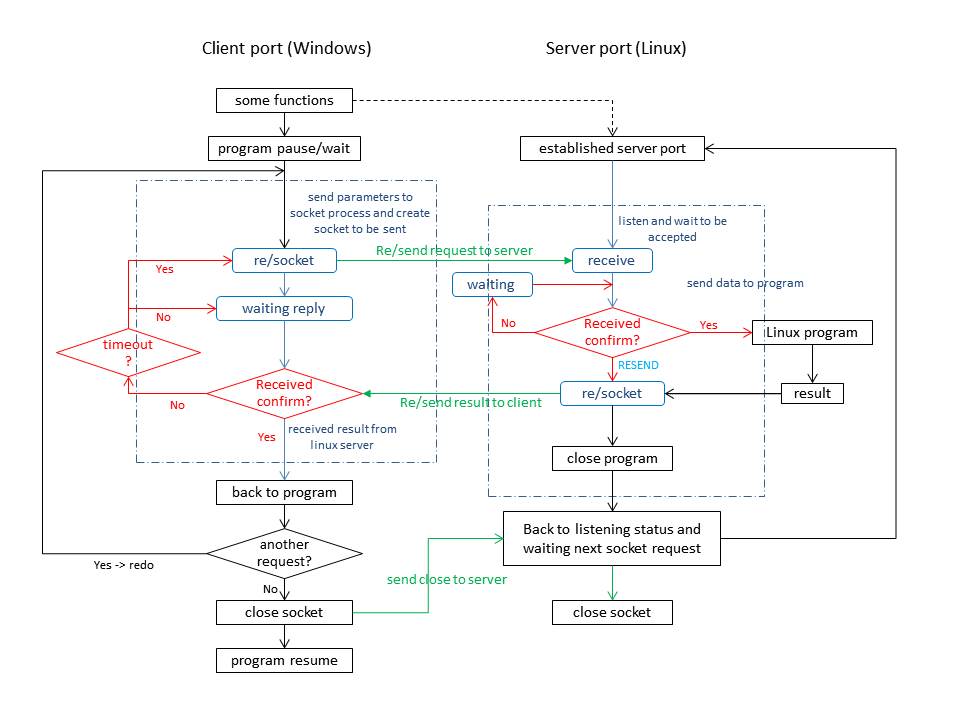


figure 1 – TCL Test Architecture Block Diagram

## 2.1.1 Server Port (Linux)

## Linux server port will be used to receive data message from window client port and run corresponding programs. Once linux port received data, some program will be started and use these data as an input value. When program has been finished, the return value will be sent to client port by socket, then close this connection.

## First of all, the establishment of server cannot be call by Windows client port, it should be established manually by source “server.tcl”. Or the server port can be established by windows program, need to be fixed in future.

## Once the server port is established, the Linux port will be in accepting status. Linux simulation will be activated when server port received any data from Windows client, then original program will be suspended until the Linux server returns a simulation result. After the simulation result sent to the client port, Linux server port will back to accepting status waiting next data message from client.

## If server port receive a “close” request from client, then close server port.

## 2.1.2 Client Port (Windows)

## Windows client port will be used to send connect requests or data messages to Linux server port and run corresponding programs. However, the client port is established by some functions or programs in Windows system and pause the original program once the data message binded with socket. For instance, windows validation program A needs a result of Linux simulation program B, then A will bind data with socket send to B, A then be suspended and waiting the return from server. When A receive the feedback from B, the simulation results, A then be resumed and continue doing its program.

## The client port will be closed after each connection or message request and can be established by the next request. If there is no more request, a close port message will be sent by client to server, then the Windows client port will be closed.

## 2.2 TCL socket mechanism

## 2.2.1 Despription

## As far as we know, socket communication mechanism is being wildly used in industrial and academic. In tools command launguage, socket programming is also follow the standard principle of commonsense except some syntax difference. The schematic diagram shown as following in figure 2.

## C:\Users\Administrator\Desktop\Windows-Linux Communication.jpg

## figure 2 – socket communication simple diagram

## Firstly, we establish server port under Linux system. The server port will start listening status and waiting connection requests from client port. Once server received such connection request, the communication has established through 3-way handshake protocol and server port will goto accept status. When server port staying in accept status, the server port is temporarly suspended until a new request sent from client port.

## Secondly, the message from client port will be used to send to some simulation program in Linux system.If server port has not received previous message, the client port will resend message to server before timeout. After that program finished simulation, the simulation results will be binded to socket and send back to client.

## Then, the original program in Windows port will decide whether close soket or not. If there are some other parameters need to be send to server, then Windows port do not have to send close port message, otherwise, a close message will close client and server port. The original program will resume when all needed results received.

## .

## 2.2.2 “user.tcl” and “client.tcl”

## In this project, there are two TCL files can be used as windows client port. The “client.tcl” file implemented simple socket communication protocol and totally follow the common socket structure, so it is easy to further reuse by adding new functions. The “user.tcl” is a well-functional TCL file for implementing continuous socket communication between Linux server and Windows client.