

# The Socket Programming and Software Design for Communication Based on Client/Server

Ming Xue

Changchun Institute of Technology  
Changchun, 130012, China  
limning@sina.com

Changjun Zhu

HeBei University of Engineering  
Handan, 056038, China  
christorf@126.com

**Abstract**—This paper introduces the application of the client/server(C/S) mode ,the concept and the programming principle of the socket based on C/S. The method of software design for the communication between the client ,server-process using the socket mechanism is mainly analyzed, and gives examples of connection-oriented service program. The transmission layer can provide connection-oriented to use TCP protocol, connectionless-oriented to use UDP protocol. There are two different kinds of services with different kinds of sockets. Only by understanding the characters of TCP protocol and UDP protocol, the service provided by the two protocols for application, can we know what deal with in these protocols, what need to deal with in application, can we easier compile vigorously and healthily, highly efficient client service program.

**Keywords**— C/S; socket ; communication ;TCP ;UDP

## I. INTRODUCTION

The computer network is the product of the combination between computer technology and communication technology. With the development of the computer network, application of the computer network has been applied to social and economic life in all aspects, for example, Bank teller, point-of-terminal, the verification of check and invoice ,etc. The design and development of network software is more and more popular, the most of system software (such as operating system UNIX, Linux, Windows versions ,etc) and application software are online versions at present. It is very important to master the network programming principle and method for designing and developing web applications.

## II. CLIENT/ SERVER APPLICATION

### A. Client/Server Mode

C/S mode is an information sharing mode which is used widely in information system. The most basic factor is custom and server in C/S environment. The client usually refers to PC or workstation. It provides the terminal client with very friendly interface, for example, Microsoft Windows and so on. The server provides the client with a group of users sharing the service program. The database server is the most common one. It makes many clients share the same access to sources of information.

Besides the client and server, the network system is the third basic factor to compose of C/S mode. Client, server computing is distributed computing. User, applications and resources are distributed, is used to

response the request of the actual business, and they can be linked through LAN, WAN or Internet .

### B. Client/Server System Structure

The core of C/S system structure is distribution of task-level application between client and server. The chart 1 gives out the mode's the general situation. The basis of exchange is communication software between client and server. The example of this kind software is TCP/IP. The main task which supports these softwares (communication software and operating system) is to provide a basic structure for distributed applications.

An essential element is a way to deal with that administrators regard the system as a whole. It can make C/S environment success. Therefore, it is very important to design client-side user interface. In the most of client/server system, its emphasis on providing easy-to-use, easy-to-learn, the powerful and flexible graphical user interface (GUI) .C/S system structure as Fig. 1.

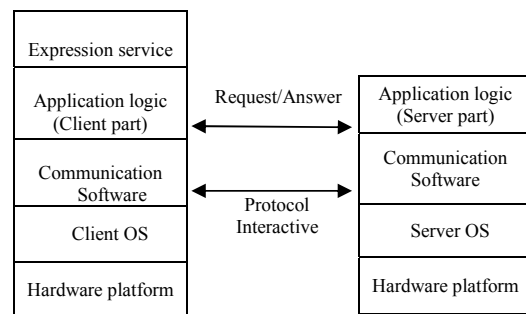


Figure 1. Client /server system structure

## III. SOCKETS

### A. Socket Interface

C/S structure allow to share information and resources between systems, for example, files, disk space, processors and peripherals can collaborate and deliver message inevitably between many processors. Some of processors are running in the client , the others are running in the server. There must be a kind of mechanism which makes every one of processor get to know the process of running another machine's network address when the two processors are communicating through the network.

The socket provides simple programming interface which has close connection with single and process communication within multi-machine. The purpose of

socket is to provide a general method IPC in the transport layer, whether these processes running on the same machine. Network socket is structured as follows:

```
#include <netinet/in.h>
Struct sockaddr_in {
    Sa_family_t sin_family; /* internet address family */
    In_port_t sin_port; /* port number */
    Struct in_addr sin_addr; /* holds the IP address */
    Unsigned char sin_zero[8]; /* filling */
};
```

### B. The Principle of Socket Communication

Socket is a kind of abstract data structure provided by operating system, it is used to build an access in the absence of correlation between the process of sending to receive messages. Socket has been providing communications endpoint process in fact and has been accomplishing the data transmission through the endpoint.

First, the socket function call attendant to create a type of Socket, then bind socket into client-side through bind function. Attendant set listening attentively to the formation the length through transferring the listen function to make a prepare from client client's requests. Attendant transfers the accept function to start in the port which binding to listen attentively comes from the connection request which the client carries.

Client-side build a type of socket through socket function(it should be similar with the type of client-side's socket). Then connect function call to the attendant where the host issued a connection request. It need to designate service where the host IP address and socket, as well as other relevant information when you link it .

In order to make communication between the machine process come true, we can use socket to achieve.

The following code lists the defined in header files UNIX domain socket address structure:

```
Struct sockaddr_un {
    Unit8_t sun_len;
    Sa_family_t sun_family; /*AF_LOCAL*/
    Char sun_path[104]; /*null-terminated
pathname*/
};
```

### C. Two Main Operations of Sockets

Sockets use connection-oriented and connectionless-oriented manner mainly, which the two manners use TCP and UDP protocol respective.

#### 1) Connection-oriented service

Connection is a combination which two of peer entities to conduct data communication. There must be connection establishment before connection-oriented service exchanges data. It will release connection when data transmission is sign-off. There are three steps during data transmission in connection-oriented service: (1) connection establishment (2) data transmission (3) release connection

To view from the three stages of Connection-oriented services ,connection is like a pipeline, sender sends messages from one side successively, recipient receive messages from another side in the same order successively. This type of connection is called virtual circuit. It can avoid to missing messages, duplicating and disorder. Using this connection, it can avoid to establishing connection and

releasing connection if two uses communicate with each other frequently.

The operate of connection-oriented use TCP protocol. TCP provides a unfailing connection-oriented byte stream transport layer service, TCP packs up the user's data into message segment; it activates a time controller after sending data; confirming the receiving data from the side of communication, and reordering the scrambled data, discarding the reduplicative data; TCP provides flow control from end to end, calculates and verifies a completive end to end check.

At present, many of popular network application such as Telnet, FTP, Rlogin and SMTP use TCP.

#### 2) Connectionless-oriented service

Under the connectionless service situation, it is no need to establish connection in advance between two entity communications .The connectionless operate uses datagram protocol. One datagram is a free-running cell, it contains all of the mailing messages. It just delivers datagram simply, which can ' t avoid the messages of missing, duplicating and disordering. Because holistic source address and destination address must be included in every message, the cost is higher than others. Connectionless operate is fast-speed and efficient, but data security is bad. Datagram is a basic service way, we can use this way if message has no relation with the order of messages arriving.

Connectionless-oriented operate use UDP protocol. UDP is a simple transport layer protocol faced datagram: every output function in the process just bring an UDP datagram, which results in an IP datagram sending. The application writes a datagram to UDP which encapsulates into IPv4 or IPv6 datagram, then sends to the destination address. However, UDP datagram can not land the destination address sometimes. The problem that will meet in using UDP program is unreliable. If you want to assure a datagram of arriving the destination address, it must build relevant feature in application, it includes: the assurance from another side, over time, retransmission and so on.

### D. The Example of Connection-Oriented Service.

In this article, we discuss the writing way of signal procedure, however, the writing way of UDP signal procedure is similar with it.

Herder give out the title files which can be used TCP and UDP.inet.h

```
/*
 *definitions for TCP and UDP client/server programs
 */
# include <stdio.h>
# include<sys/types.h>
# include<sys/socket.h>
# include<netinet/in.h>
# include<arpa/inet.h>
# define SERV_UDP_PORT 6000
# define SERV_TCP_PORT 6000
# define SERV_HOST_ADDR " 1 92.43.235.6 "
/*host addr for server*/
Char *pname;
COM port is optional here, but it must be greater than
1023, it has better greater than 5000, and it can ' t be
```

conflict with other TCP port. SERV\_HOST\_ADDR constant is corresponding with the host computer users.

```

1) The follow is service program:
/*Example of server using TCP protocol.*/
#include "inet.h"
Main(argc,argv)
Int argc;
Char *argv[];
Int sockfd,newsockfd,cliden,Childpid;
Struct sockaddf_in cli_addr ,serv_addr;
Pname=argv[0];
/*Open a TCP socket(an Internet Stream socket).*/
If((sockfd=socket (AF_INET ,SOCK_STREAM ,
0))<0)
Err_dump ( "server:can not open stream socket " );
/*Bind our local address so that the client can send to
us .*/
Bzero((Char *)&serv_addr,sizeof(serv_addr ));
Serv_addr.sin_family=AF_INET;
Ser_addr.sin_addr.s_addr=htonl(INADDR_ANY)
Serv_addr.sin_port=htons(SERV_TCP_PORT);
If (bind(sockfd ,(struct sockaddr *)&serv_addr ,sizeof
(serv_addr))<0)
Err_dump ( "server:can not bind local address " );
Listen (sockfd , 5);
For ( ; ; ){
/* Wait for a connection fromm a client process,
This is an example of a concurrent server. */
Clilen = sizeof (cli_addr);
Newsockfd=accept (sockfd, (struct sockaddr
*)&cli_addr,&clilen );
If(newsockfd <0)
Err_dump( "server: accept error " );
If((childpid =fork ())<0)
Err_dump( "server: fork error " );
Else if (childpid ==0){ /* child process */
Close (sockfd) ; /* close original socket */
Str_echo (newsockfd) ; /*process the request */
Exit (0);
Close (newsockfd) ; /* parent process */
}
}

```

Constant INADDR\_ANY is the Internet address which is designated by bind service, it tells the system that if there are some connections , it will accept any one of the system connection in the internet interface.

2) The follow is client program:

```

/* Example of client using TCP protocol.*/
#include "inet.h"
Main (argc, argv )
Int argc ; ,
Char * argv [];
{
Int sockfd ;
Struct sockaddr_in serv_addr;
Panme=argv[0];
/* Fill in the structure "serv_addr" with the address
of the
* server that we want to connect with */
Bzero ((char *) &serv_addr, sizeof (serv_addr));
Serv_addr.sin_family=AF_INET;
Serv_addr.sin_addr.s_addr=inet_addr(SERV_HOST_
ADDR);
Serv_addr.sin_port=htons (SERV_TCP_PORT);
/* Open a TCP socket (an Internet stream socket). */
If (sockfd=socket(AF_INET, SOCK_STREAM, 0)<0)
Err-sys( "client :can not open stream socket " );
/* Connect to the server. */
If (connect (sockfd , (struct sockaddr *)& serv_addr
,sizeof (serv_addr))<0
Err_sys( "client:cannot connect to server" );
Str_cli (Stdin , sockfd) ; /* do it all */
Close (sockfd) ;
Exit (0);
}

```

#### IV. CONCLUSION

C/S mode is an information sharing mode which is used widely in information system. The core of client /server system structure is distribution of task-level application between client and server. In order to make it easier to compile efficient client service program., we need a deep understanding of the TCP protocol and UDP protocol on the transport layer.

#### REFERENCES

- [1] Huang weiming Automated computer of Chengdu Institute Meteorology Journal, 1996, 11 ,43.
- [2] Liang Hu, Li Qiang, Network Programming Jilin University Press, 2002.
- [3] Willian Stalling,Data and Computer Communications;Fifth Edition Prentice-Hall Internation Inc. 1999
- [4] Jasperreport Official Website , <http://Jasperreports.sourceforge.net>.