

## 实验 2.7 socket 编程\*

### 【实验目的】

- 了解 TCP 服务器/客户机通信的基本过程

### 【实验过程】

#### 1 TCP 服务器代码

实例：TCP 服务器程序 server.c

```
#pragma comment(lib,"ws2_32.lib")
#include <Winsock2.h>
#include <stdio.h>
#include <stdlib.h>
#define DEFAULT_PORT 5050                                //服务端默认端口

int main(int argc, char* argv[])
{
    int      iPort = DEFAULT_PORT;
    WSADATA  wsaData;
    SOCKET   sListen,sAccept;
    int      iLen;                                       //客户机地址长度
    int      iSend;                                       //发送数据长度
    char      buf[] = "I am a server";                 //要发送给客户的信息
    struct sockaddr_in ser,cli;                          //服务器和客户的地址
    if(WSAStartup(MAKEWORD(2,2),&wsaData)!=0)
    {
        printf("Failed to load Winsock.\n");           //Winsock 初始化错误
        return -1;
    }
    sListen = socket(AF_INET,SOCK_STREAM,0);             //创建服务器端套接字
    if(sListen == INVALID_SOCKET)
    {
        printf("socket() Failed: %d\n",WSAGetLastError());
        return -1;
    }

    //以下初始化服务器端地址
    ser.sin_family = AF_INET;                           //使用 IP 地址族
    ser.sin_port = htons(iPort);                        //主机序端口号转换为网络字节序端口号
    ser.sin_addr.s_addr = htonl(INADDR_ANY);            //主机序 IP 地址转换为网络字节序主机地址
```

```

//使用系统指定的 IP 地址 INADDR_ANY
if(bind(sListen,(LPSOCKADDR)&ser,sizeof(ser)) == SOCKET_ERROR) //套接定与地址的绑定
{
    printf("bind() Failed: %d\n",WSAGetLastError());
    return -1;
}
if(listen(sListen,5) == SOCKET_ERROR) //进入监听状态
{
    printf("lisiten() Failed: %d\n",WSAGetLastError());
    return -1;
}
iLen = sizeof(cli); //初始化客户端地址长度参数

while(1) //进入循环等待客户的连接请求
{
    sAccept = accept(sListen,(struct sockaddr *)&cli,&iLen);
    if(sAccept == INVALID_SOCKET)
    {
        printf("accept() Failed: %d\n",WSAGetLastError());
        return -1;
    }
    printf("Accepted client IP:[%s],port:[%d]\n",inet_ntoa(cli.sin_addr),ntohs(cli.sin_port));
    //输出客户端 IP 地址和端口号
    iSend = send(sAccept,buf,sizeof(buf),0); //给客户端发送信息
    if(iSend == SOCKET_ERROR) //错误处理
    {
        printf("send() Failed: %d\n",WSAGetLastError());
        break;
    }
    else if(iSend == 0)
    {
        break;
    }
    else
    {
        printf("send() byte: %d\n",iSend); //输出发送成功字节数
    }
    closesocket(sAccept);
}
closesocket(sListen); //关闭 socket
WSACleanup(); //输出发送成功字节数
return 0;
}

```

## 2 TCP 客户机程序

实例：TCP 客户机程序 client.c

```
#pragma comment(lib,"ws2_32.lib")
#include <Winsock2.h>
#include <stdio.h>
#include <stdlib.h>
#define DATA_BUFFER 1024 //默认缓冲区大小

int main(int argc, char * argv[])
{
    WSADATA wsaData;
    SOCKET sClient;
    int iPort = 5050;
    int iLen; //从服务器端接收的数据长度
    char buf[DATA_BUFFER]; //接收缓冲区
    struct sockaddr_in ser; //服务器端地址

    if(argc<2) //判断参数输入是否正确：client [Server IP]
    {
        printf("Usage: client [server IP address]\n"); //命令行提示
        return -1;
    }
    memset(buf,0,sizeof(buf)); //初始化接收缓冲区
    if(WSAStartup(MAKEWORD(2,2),&wsaData)!=0)
    {
        printf("Failed to load Winsock.\n"); //Winsock 初始化错误
        return -1;
    }

    ser.sin_family = AF_INET; //初始化服务器地址信息
    ser.sin_port = htons(iPort); //端口转换为网络字节序
    ser.sin_addr.s_addr = inet_addr(argv[1]); //IP 地址转换为网络字节序

    sClient = socket(AF_INET,SOCK_STREAM,0); //创建客户端流式套接字
    if(sClient == INVALID_SOCKET)
    {
        printf("socket() Failed: %d\n",WSAGetLastError());
        return -1;
    }

    //请求与服务器端建立 TCP 连接
    if(connect(sClient,(struct sockaddr *)&ser,sizeof(ser)) == INVALID_SOCKET)
    {
```

```

        printf("connect() Failed: %d\n",WSAGetLastError());
        return -1;
    }
    else
    {
        iLen = recv(sClient,buf,sizeof(buf),0);          //从服务器端接收数据
        if(iLen == 0)
            return -1;
        else if(iLen == SOCKET_ERROR)
        {
            printf("recv() Failed: %d\n",WSAGetLastError());
            return -1;
        }
        else
            printf("recv() data from server: %s\n",buf); //输出接收数据
    }
    closesocket(sClient);                                //关闭 socket
    WSACleanup();
    return 0;
}

```

### 3 server 和 client 程序的编译

首先安装 Windows 平台的 C 语言开发环境 CFree5.0。

在 CFree5.0 菜单的工程选项中新建一个工程。工程类型为 Win32 控制台程序，工程名称命名为 server，程序类型为空的程序，构建配置选择默认的 mingw5。

接着在构建菜单的构建选项中选择连接，在连接库中添加“ws2\_32”。

然后添加一个空白文件，将 server.c 代码复制进入这个文件，保存并命名为 server.c。

最后构建并运行这个项目，编译生成 server.exe。

以同样的方法新建 client 工程，编译生成 client.exe。

在 CFree5.0 的消息窗口可以观察到构建项目编译生成的可执行文件放置目录。

### 4 TCP 客户机和服务器的一次通信

打开两个命令行窗口，分别进入步骤 3 编译生成的可执行文件目录，首先使用 ipconfig 命令查看主机的 IP 地址，然后启动 server.exe，最后启动 client.exe 程序与服务器通信。条件允许的情况下可以在一台主机运行 server.exe，另外一台主机运行客户机与服务器进行通信。

#### 实例：client 和 server 的一次通信

```

C:\>server
Accepted client IP:[192.168.1.101], port:[5716]
send() byte: 14

```

~首先启动 server

~显示客户机信息

~显示发送的字节数

```

C:\>client 192.168.1.101
recv() data from server: I am a server.

```

~启动 client，本机通信可以使用 127.0.0.1

~显示接收的信息