



第3章 网络数据传输程序



实训目的

- (1) 理解进程通信的原理及通信过程。
- (2) 掌握基本的网络编程方法。



实训任务

- (1) 学习应用 C 语言与 WinSock2 进行简单的无连接的网络程序设计。
- (2) 学习应用 C 语言与 WinSock2 进行简单的面向连接的网络程序设计。
- (3) 实现网络数据传输。



实训设备

两台装有 Microsoft Visual C++ 6.0 的 PC 机。



实训步骤

1. 服务端的建立

(1) 打开 Microsoft Visual C++ 6.0, 依次选择“文件”→“新建”→“工程”→ Win32 Console Application, 并将工程命名为 Socketserver, 如图 3-1 所示。单击“确定”按钮, 然后选择“一个简单的空工程”, 并单击“完成”按钮, 开始进行服务端代码的编写。



图 3-1 Socketserver 的建立



(2) 依照代码清单 3-1 进行服务端的编辑,界面如图 3-2 所示。

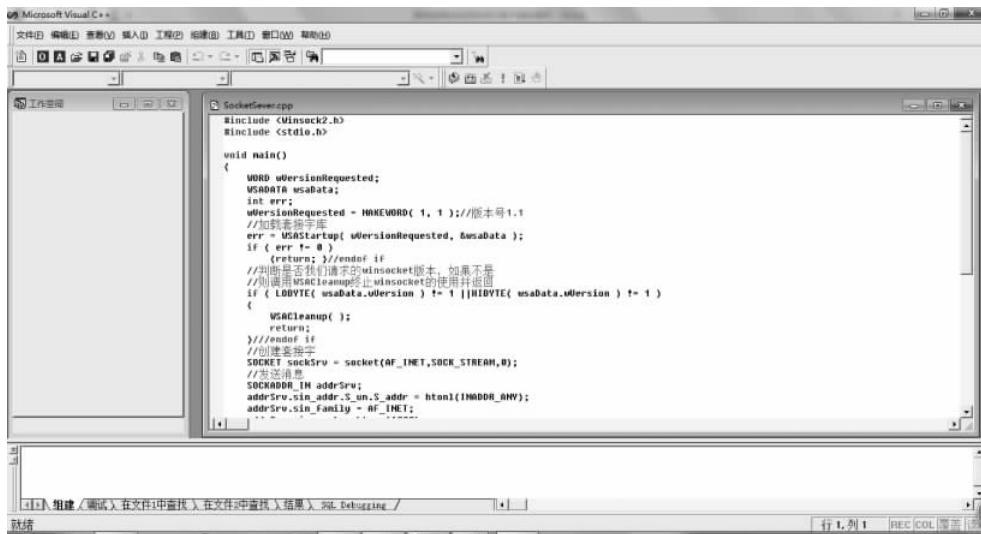


图 3-2 服务端代码示意

代码清单 3-1:

```
#include <stdio.h>
#include <Winsock2.h>
void main()
{
    WORD wVersionRequested;
    WSADATA wsaData;
    int err;
    wVersionRequested = MAKEWORD( 1, 1 );
    err = WSAStartup( wVersionRequested, &wsaData );
    if ( err != 0 ) {
        return;
    }
    if ( LOBYTE( wsaData.wVersion ) != 1 ||
        HIBYTE( wsaData.wVersion ) != 1 ) {
        WSACleanup( );
        return;
    }
    SOCKET sockSrv = socket(AF_INET, SOCK_STREAM, 0);
    SOCKADDR_IN addrSrv;
    addrSrv.sin_addr.S_un.S_addr = htonl(INADDR_ANY);
    addrSrv.sin_family = AF_INET;
    addrSrv.sin_port = htons(6000);
    bind(sockSrv, (SOCKADDR *) &addrSrv, sizeof(SOCKADDR));
    listen(sockSrv, 5);
    SOCKADDR_IN addrClient;
    int len = sizeof(SOCKADDR);
    while(1)
```



```
{
    SOCKET sockConn = accept(sockSrv, (SOCKADDR *)&addrClient, &len);
    char sendBuf[50];
    sprintf(sendBuf, "Welcome %sto
here!", inet_ntoa(addrClient.sin_addr));
    send(sockConn, sendBuf, strlen(sendBuf) + 1, 0);
    char recvBuf[50];
    recv(sockConn, recvBuf, 50, 0);
    printf(" %s\n", recvBuf);
    closesocket(sockConn);
}
}
```

(3) 选择“工程”→“设置”→“连接”命令,在“分类”下拉列表框中选择“常规”,在“对象/库模块”中添加 ws2_32.lib,如图 3-3 所示。单击“确定”按钮,调试通过后运行程序,结果如图 3-4 所示。



图 3-3 lib 的添加

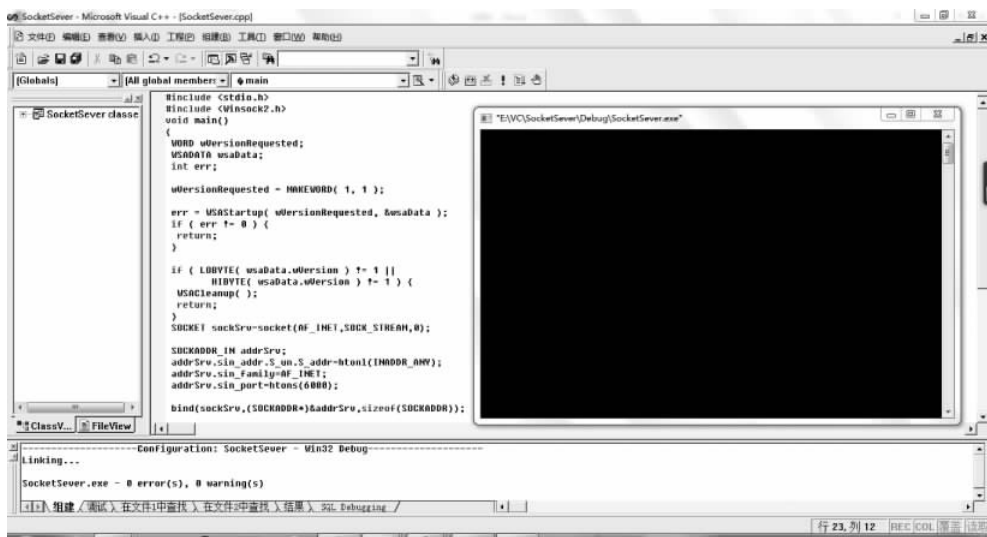


图 3-4 服务端运行示意



2. 客户端的建立

(1) 打开 Microsoft Visual C++ 6.0, 选择“文件”→“新建”→“工程”→Win32 Console Application, 并将工程命名为 SocketClient 如图 3-5 所示。单击“确定”按钮, 选择“一个简单的空工程”, 并单击“完成”按钮, 然后编写客户端的代码。



图 3-5 客户端的建立

(2) 按照代码清单 3-2 进行编写。

代码清单 3-2:

```
#include <stdio.h>
#include <Winsock2.h>
void main()
{
    WORD wVersionRequested;
    WSADATA wsaData;
    int err;
    wVersionRequested = MAKEWORD( 1, 1 );
    err = WSStartup( wVersionRequested, &wsaData );
    if ( err != 0 ) {
        return;
    }
    if ( LOBYTE( wsaData.wVersion ) != 1 ||
        HIBYTE( wsaData.wVersion ) != 1 ) {
        WSACleanup( );
        return;
    }
    SOCKET sockClient = socket(AF_INET, SOCK_STREAM, 0);
    SOCKADDR_IN addrSrv;
    addrSrv.sin_addr.S_un.S_addr = inet_addr("195.168.1.105");
    addrSrv.sin_family = AF_INET;
    addrSrv.sin_port = htons(6000);
    connect(sockClient, (SOCKADDR *) &addrSrv, sizeof(SOCKADDR));
    send(sockClient, "hello", strlen("hello") + 1, 0);
    char recvBuf[50];
```



```
recv(sockClient, recvBuf, 50, 0);
printf(" %s\n", recvBuf);
closesocket(sockClient);
WSACleanup();
}
```

(3) 选择“工程”→“设置”→“连接”命令,在“分类”下拉列表框中选择“常规”,在“对象/库模块”中添加 ws2_32.lib,并单击“确定”按钮,运行结果如图 3-6 所示。

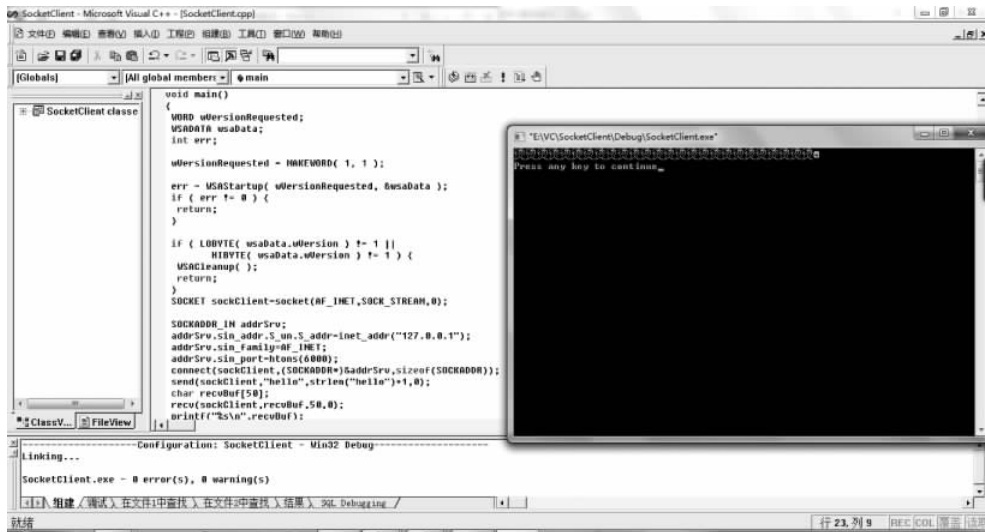


图 3-6 客户端运行示意

3. 实现服务端与客户端的交互

根据存储路径分别在 debug 文件夹中找到类型为应用程序的可执行文件,先执行客户端,再执行服务端,运行结果如图 3-7、图 3-8 所示。

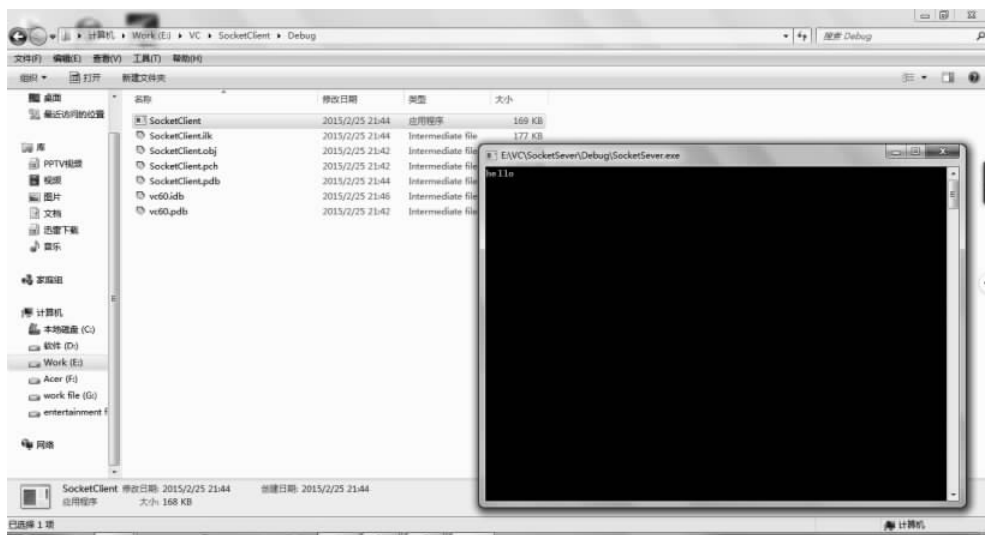


图 3-7 应用程序路径示意

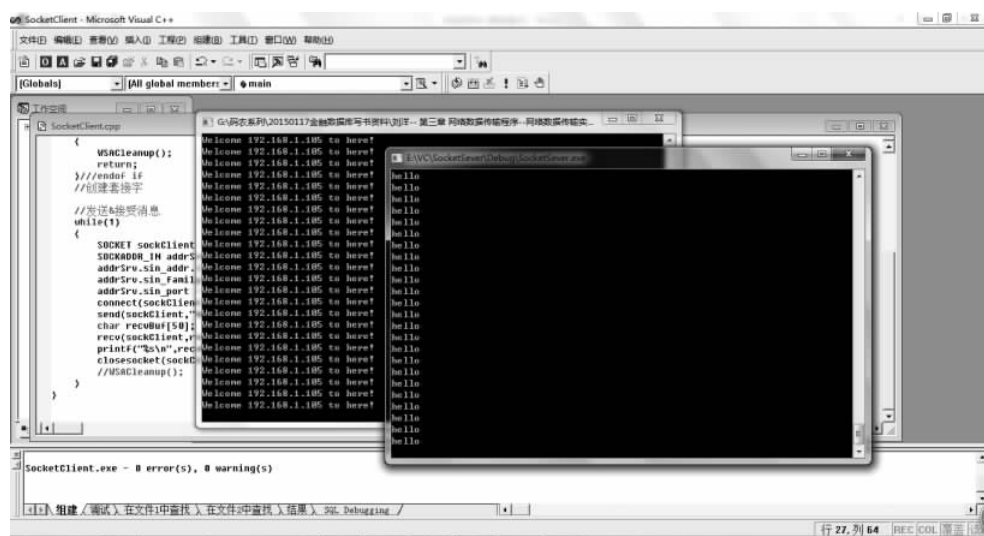


图 3-8 两端运行交互示意