

西安交通大学实验报告

课程 **计算机网络基础** 实验名称 **网络编程** 共 **22** 页
系 别 **理学院** 实 验 日 期 **2004 年 3 月 23 日**
专业班级 **信息与计算科学 01 班** 实验报告日期 **2004 年 3 月 23 日**
姓 名 **牛凌峰** 学号 **00092004** 报 告 退 发 (订 正 、 重 做)
E_mail: insisting@263.net 教师 审 批 评 分 _____

网络编程

一、实验任务

- (1) 熟悉和掌握网络编程的基本方法和步骤;
- (2) 进一步理解 client/server 交互模式;
- (3) 加深对于网络协议概念的理解以及协议的设计和实现方法;
- (4) 使用任意网络编程语言编写网络选课模拟程序, 它由 client 和 server 两部分组成, client 和 server 之间的通信基于 NCSP 协议。

二、环境

本程序使用 C++builder 6.0 开发, 图形界面窗口, 在 Windows 环境下运行。程序对于机器类型, 内存、cpu 主频, 没有特殊要求, 一般的 PC 机均可以运行服务器程序和客户机程序。

三、设计思路

1. 程序的逻辑框图如图 1 所示。
2. 所使用的数据库

采用 C++ Builder 中的 Database Desktop 创建 dBASE for Windows 数据库, 在上传的文件中可以看见名称为 course.dbf, 类型为 Microsoft Visual FoxPro Table 的数据表。这里所存放的就是已经选课学生的选课信息。

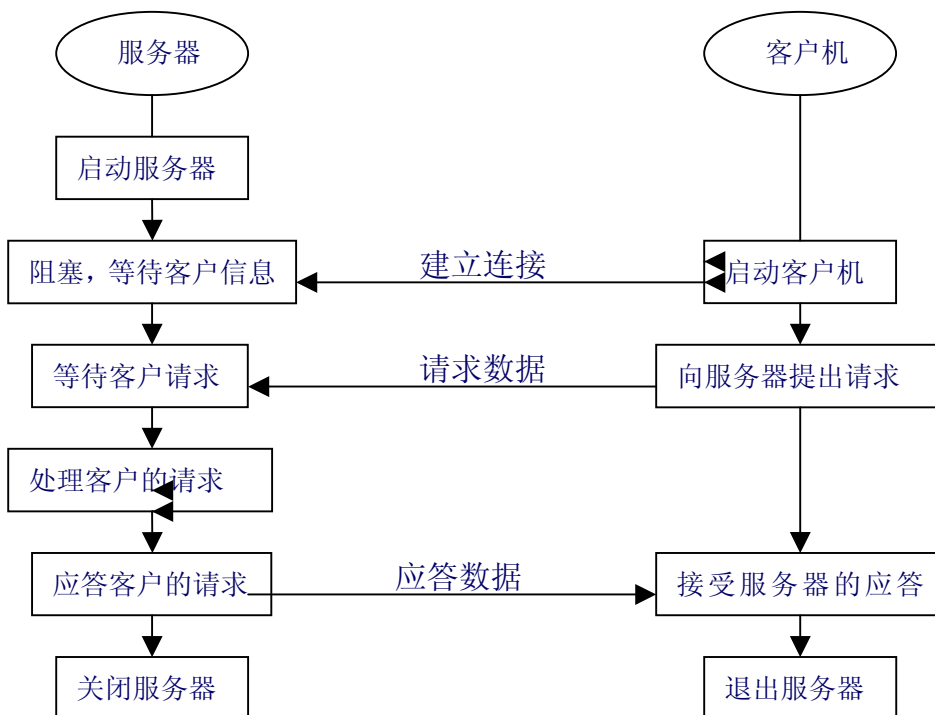


图 1 程序的逻辑框图

3. 两个窗口界面的结构和其中各个组件的主要功能

在服务器窗口中，有一个“启动”按钮、一个“关闭”按钮和一个 ServerMemo。如图 1 所示。

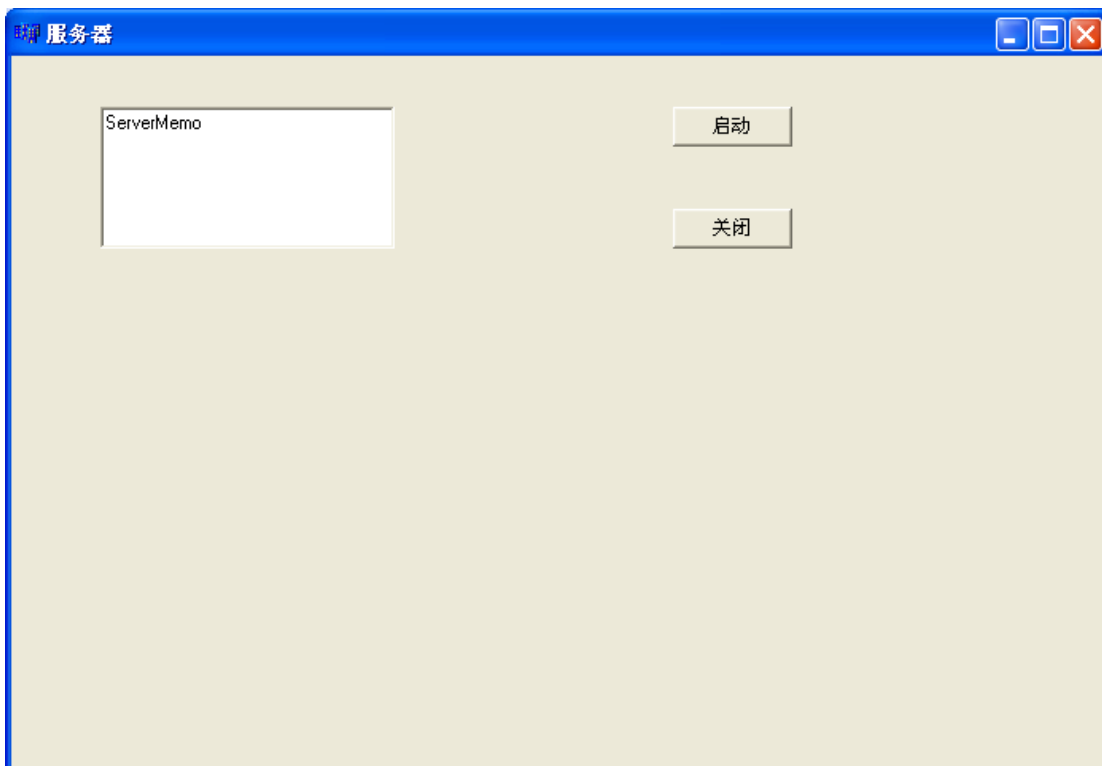


图 1 服务器窗口

点击“启动”按钮，即通过调用 StartButtonClick ()函数，打开服务器 (C++ Builder 的 ServerSocket 组件所带的各个函数就已经完成建立套接字、将套接字和本地网络地址联系在一起、使套接字做好侦听的准备、规定它的请求队列的长度、接收连接等工作)；点击“关闭”按钮即调用 CloseButtonClick()函数，关闭服务器；在 ServerMemo 中，会显示诸如：有客户登陆、客户离开、超级用户远程关闭服务器等信息，显示服务器当前所处的状态，方便管理员的管理。

在客户机窗口上，有一个“请输入您的用户名”编辑框、一个“请输入您要选的课程”编辑框、一个 ClientMemo 和八个按钮（其内容分别是“登陆”、“退出”、“我是否可以选课”、“我是否可以查询”、“我是否可以远程关闭服务器”、“选课确定”、“查询”、“远程关闭服务器”），如图 2 所示。



图 2 客户机窗口

- 1) “请输入您的用户名”编辑框，用于输入用户名；
- 2) “登陆”按钮，用于登陆到服务器；
- 3) “退出”按钮，用于用户退出服务器；
- 4) ClientMemo 用于显示服务器端发来的信息；
- 5) “我是否可以选课”按钮，用于客户询问自己是否有权选课；
- 6) “请输入您要选的课程”编辑框，用于输入有权选课的客户端输入的要选的课程；
- 7) “我是否可以查询”按钮，用于客户询问自己是否已经选程，能否即进行查询；
- 8) “查询”按钮，用于有权查询的客户端查询自己的选课情况；

9) “我是否可以远程关闭服务器”按钮，用于用户查询自己是否为超级用户；

10) “远程关闭服务器”按钮，用于超级用户远程关闭服务器，该按钮只有在核对了用户为超级用户后才可用，否则灰显；

4. 服务器与客户端的运行机制

客户机启动，在正确输入用户名以后，点击“登陆”按钮，就可以正确登陆到服务器上，这时在 ClientMemo 中会显示出服务器发送给您的欢迎信息。但是此时您不能进行选课、查询选课信息、远程关闭服务器其中任何一项的功能（因为所有相应的按钮都是灰显的，如图 2 所示），您可以点击“我是否可以选课”、“我是否可以查询”、“我是否可以远程关闭服务器”按钮，查看自己拥有的权限。服务器以客户的用户名做为关键字，在数据库中进行查询，如果在数据库中查到了该用户名，则认为该用户已经选课，不能再选课，只能查询所选课程，如果在数据库中没有查找到该用户名，则这个用户就可以进行选课（即在选课的编辑框中输入要选的课程），然后点击“确定”按钮，客户机将选课信息发送给了服务器，服务器将这条记录加到了数据库中，下次当这个用户再登陆的时候，就可以查询自己的选课情况了。点击“我是否可以远程关闭计算机”按钮，服务器将会检验所输入的用户名是否为超级用户，如果是则“远程关闭服务器”的按钮将可用，超级用户将可以远程关闭服务器，如果不是，服务器将会发送消息到客户机，告知客户他没有这个权限。总而言之，服务器将向客户发送提示信息，告诉客户他所拥有的权限，并且只有客户所拥有的权限其相应的按钮才是可用的，确保了程序的安全性。

最后，待数据传送结束后，一般的客户可以在客户端点击“退出”按钮，退出服务器，与此同时，在服务器端，将会显示客户已经离开了服务器的提示信息。

对于超级用户不但可以使用上述的方法退出服务器，还可以远程关闭服务器。在服务器关闭之前，服务器会给超级用户发送他已经关闭服务器的提示信息并且在客户端的 ClientMemo 中显示出来，同时服务器端的 ServerMemo 也会显示超级用户已经关闭服务器的提示信息。

在没有客户登陆服务器的时候，管理员可以在服务器端关闭服务器，并在服务器端的 ServerMemo 会显示服务器关闭的提示信息，如图 3 所示。

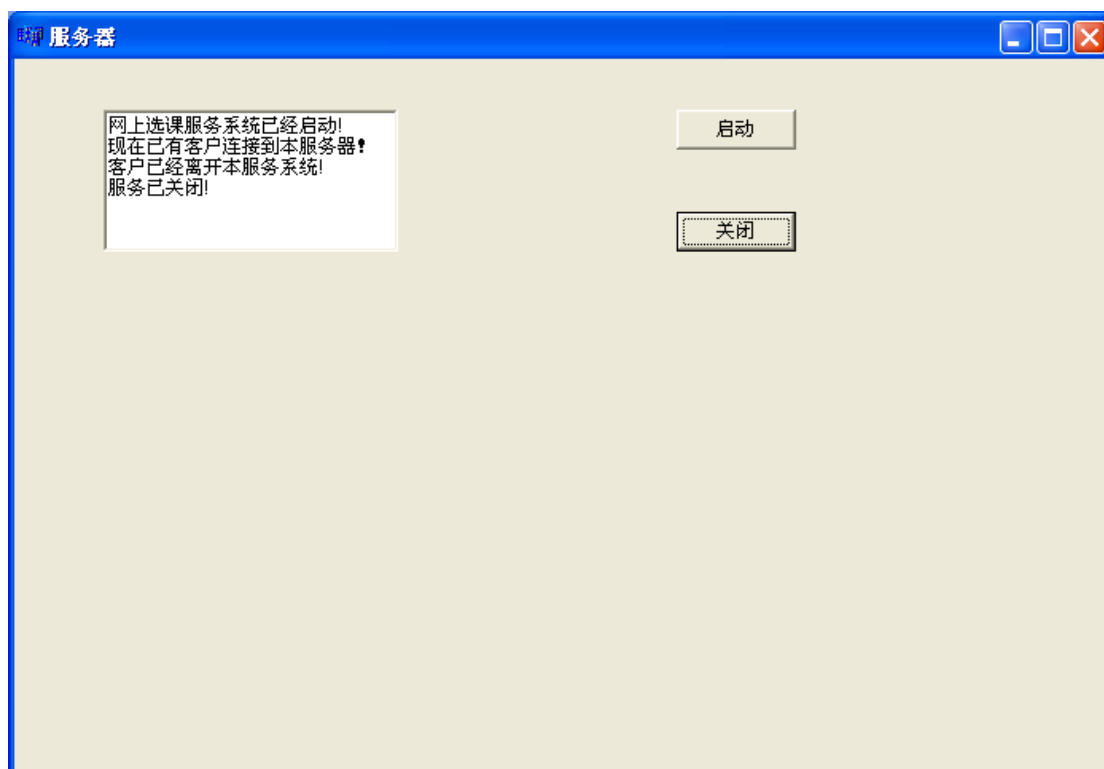


图 3

四、实验过程

实验步骤：

1. 双击 ServerProject 的图标，运行 Server 程序，如图 1 所示；
2. 点击服务器窗口上的“启动”按钮，启动服务器，ServerMemo 上显示出服务器启动的提示信息，如图 4 所示；

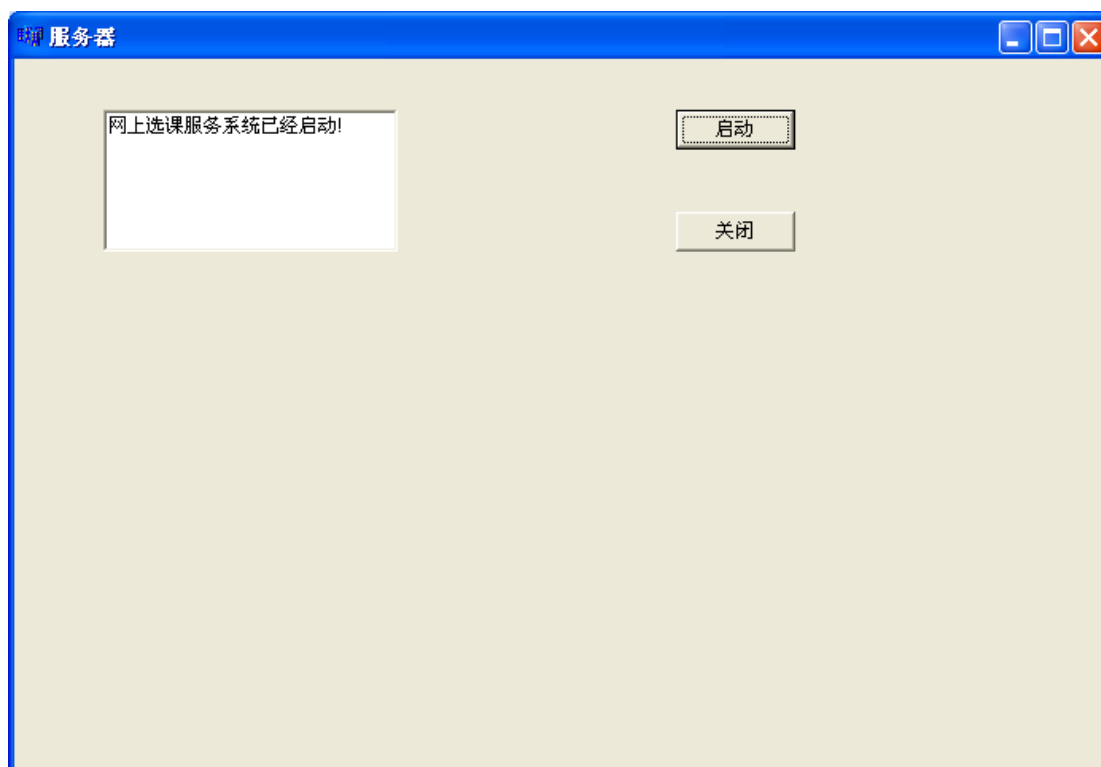


图 4

3. 双击 ClientProject 的图标，运行 Client 程序，如图 2 所示；
4. 在“输入您的用户名”编辑框中输入您的用户名，然后点击“登陆按钮”，便可以登陆到服务器上，ClientMemo 中会显示服务器发来的欢迎信息如图 5 所示；

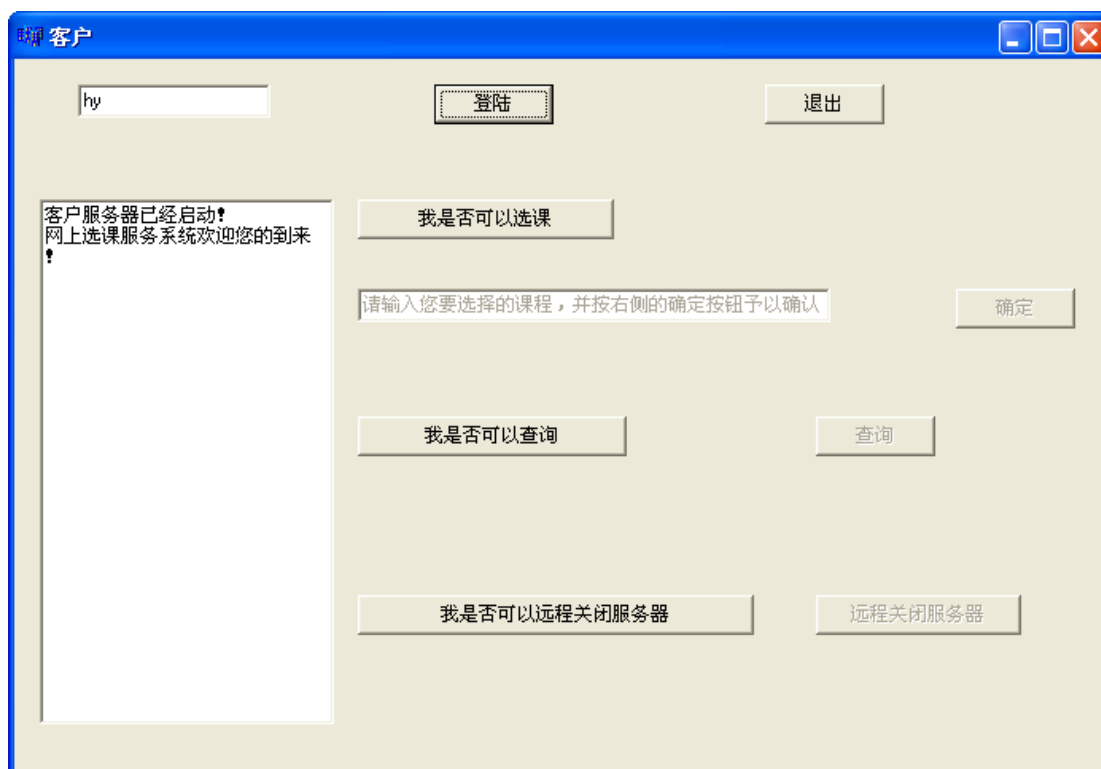


图 5

5. 点击“我是否可以查询”按钮，查询选课情况，会出现两种结果，即用户没有选课不能查询和已经选课可以查询，在 ClientMemo 中均会给出相应的提示信息，分别如图 6 和图 7 所示；

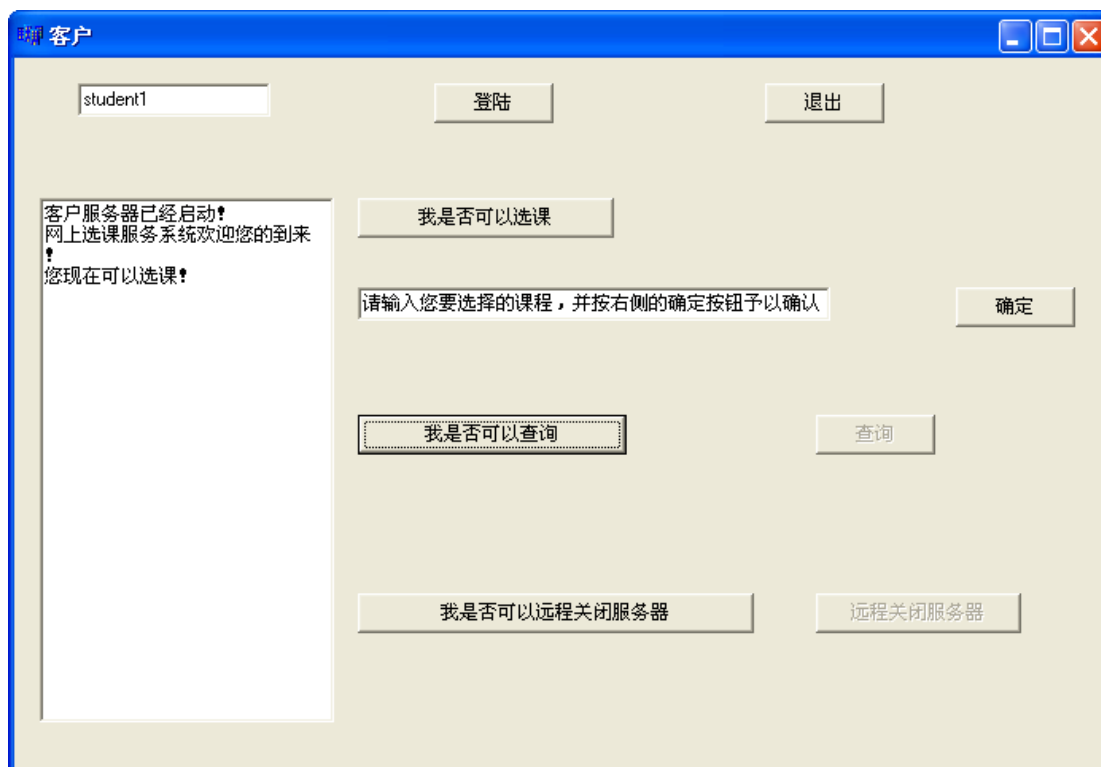


图 6

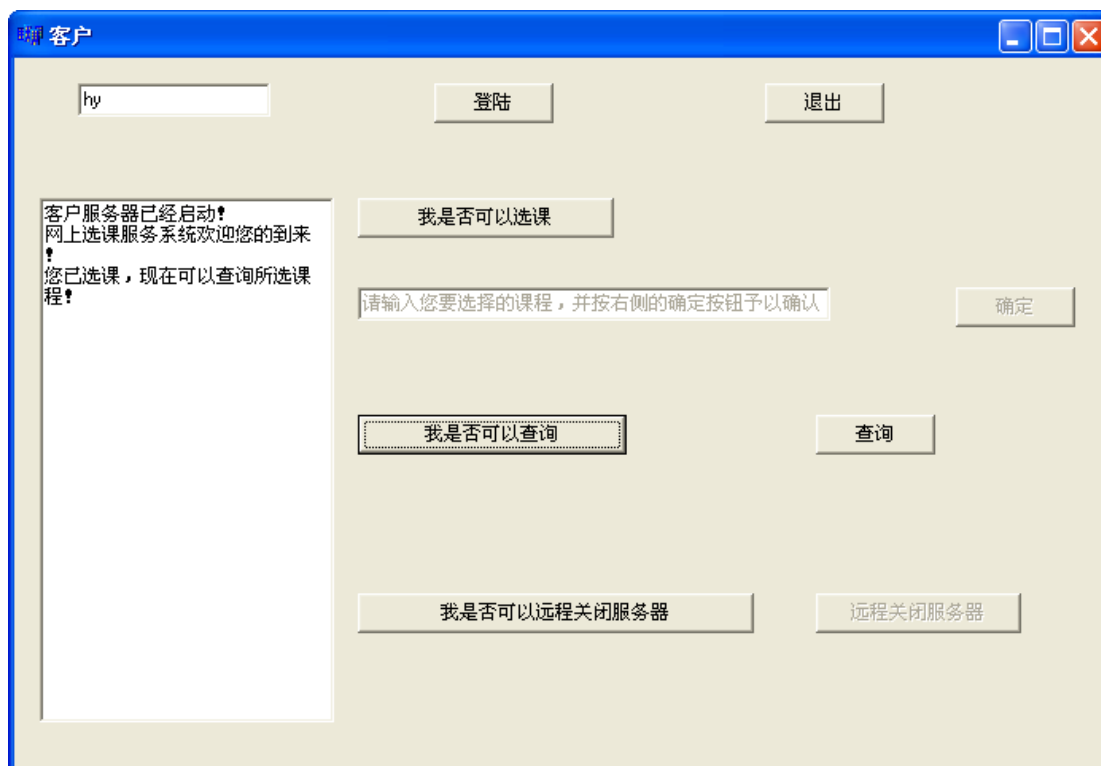


图 7

6. 点击“我是否可以选课”按钮，询问服务器客户是否可以选课，会出现两

- 种结果，即没有选课可以选课和已经选课不能再选，如图 6 和图 7 所示；
7. 在可以查询的情况下，点击“查询”按钮，服务器就会将这个客户的选课信息发送到客户端，客户端再将它显示在 ClientMemo 上，如图 8 所示；

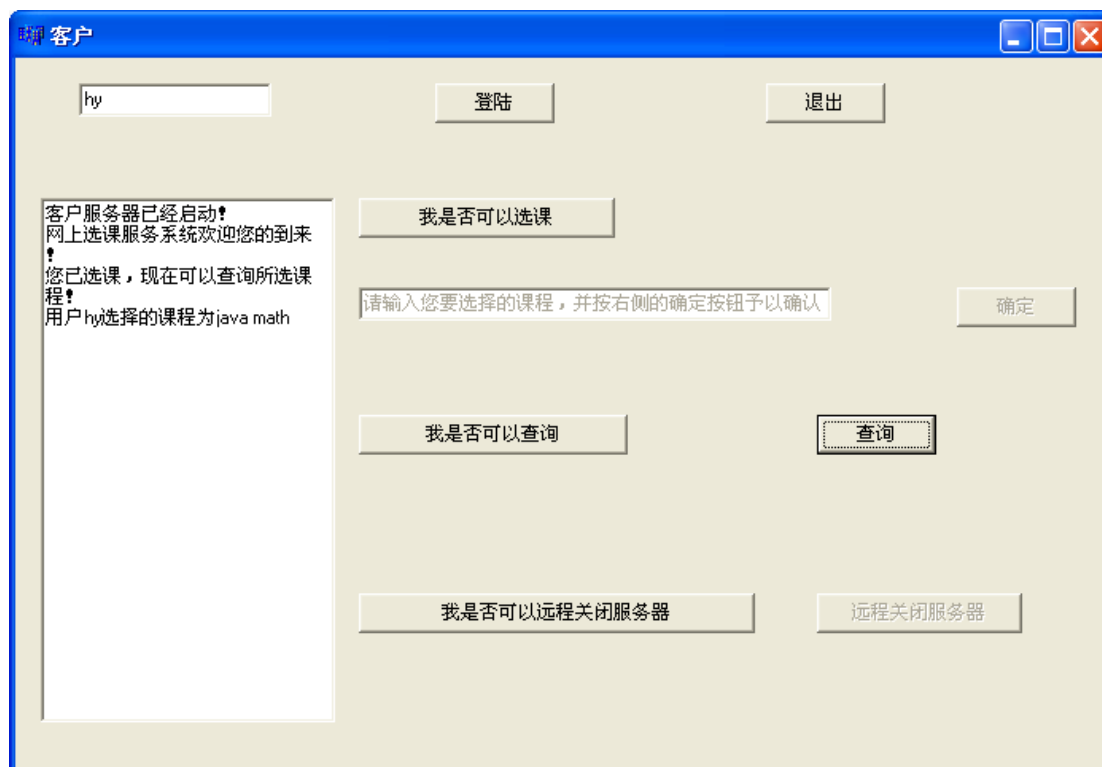


图 8

8. 在可以选课的情况下，在“输入您要选的课程”的编辑框中，输入所要选的课程，单击右边的“确定”按钮，客户端就会将这个用户的选课信息发送给服务器，服务器再将信息存入数据库，下次这个用户再登陆的时候就可以查询自己的选课情况了，如图 9 所示；

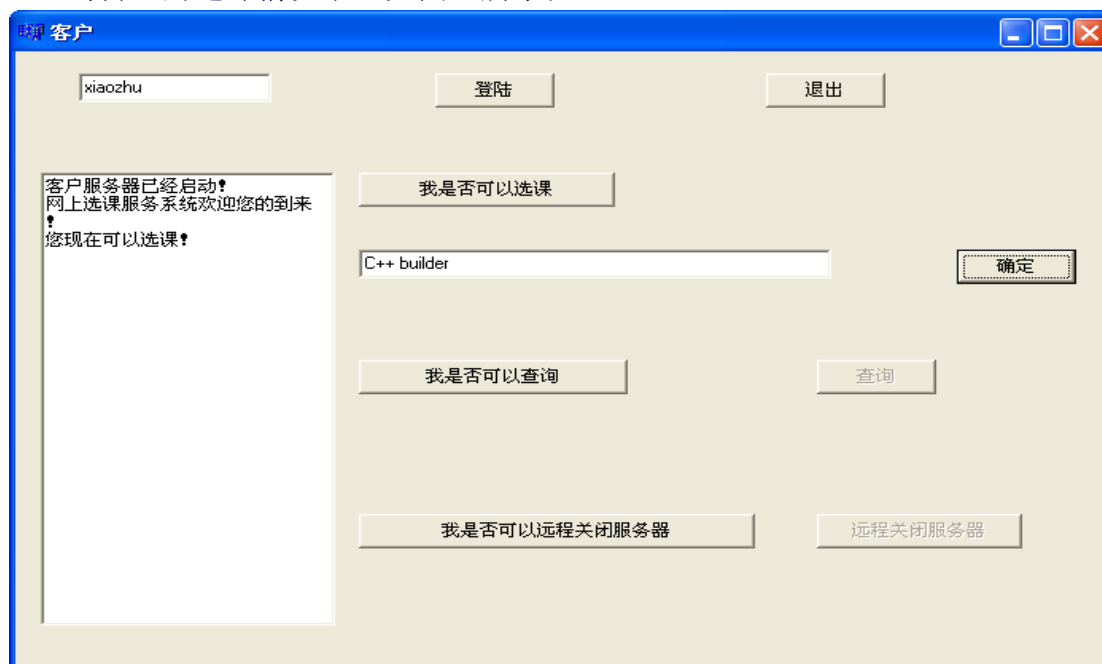


图 9

9. 单击“我是否可以远程关闭服务器”按钮，询问用户是否可以远程关闭服务器，会有两种结果，一是可以远程关闭服务器，启动“远程关闭服务器”按钮，一是非超级用户，不能远程关闭服务器，两种情况服务器均会发送提示信息给客户，并有客户机在 ClientMemo 中显示提示信息，如图 10 和图 11 所示；

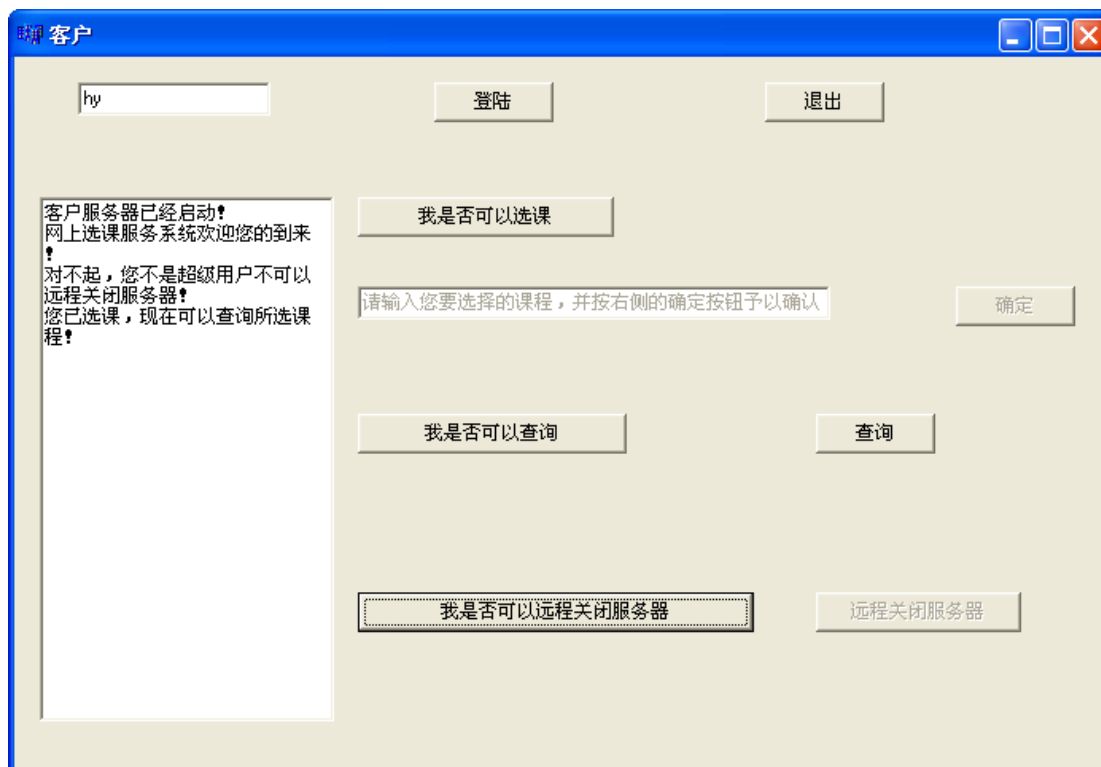


图 10

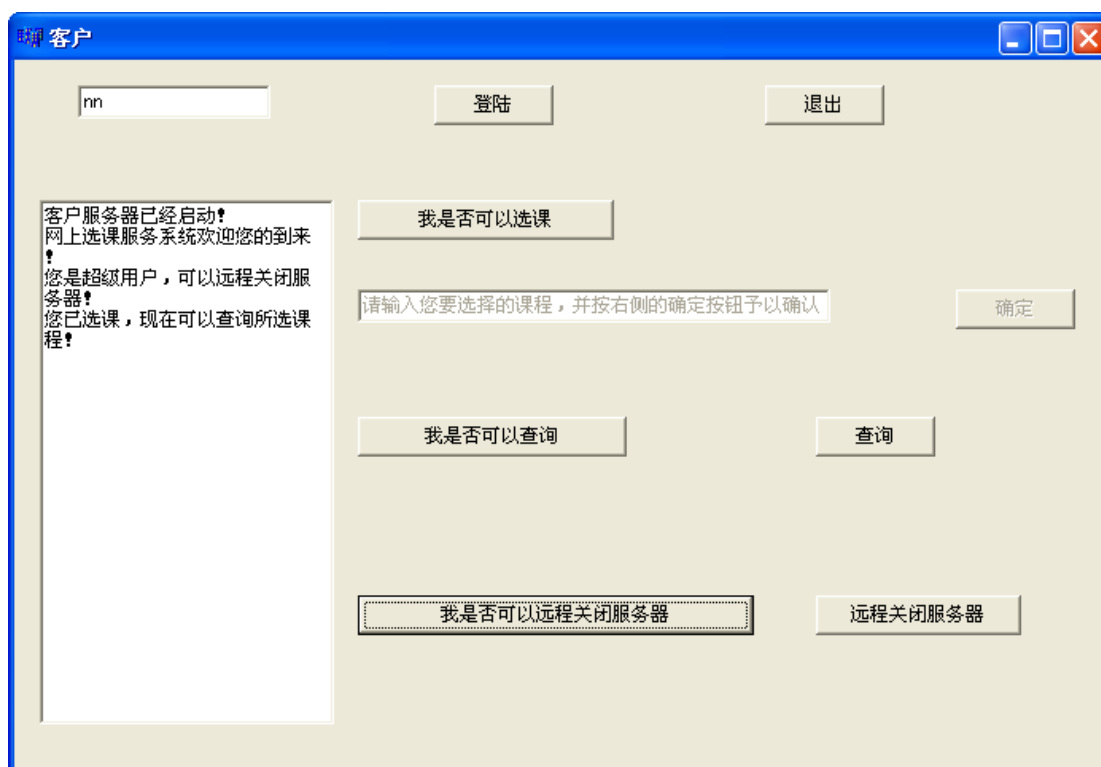


图 11

10. 超级用户关闭服务器后，服务器与客户端的情况分别如图 12 和图 13 所示；

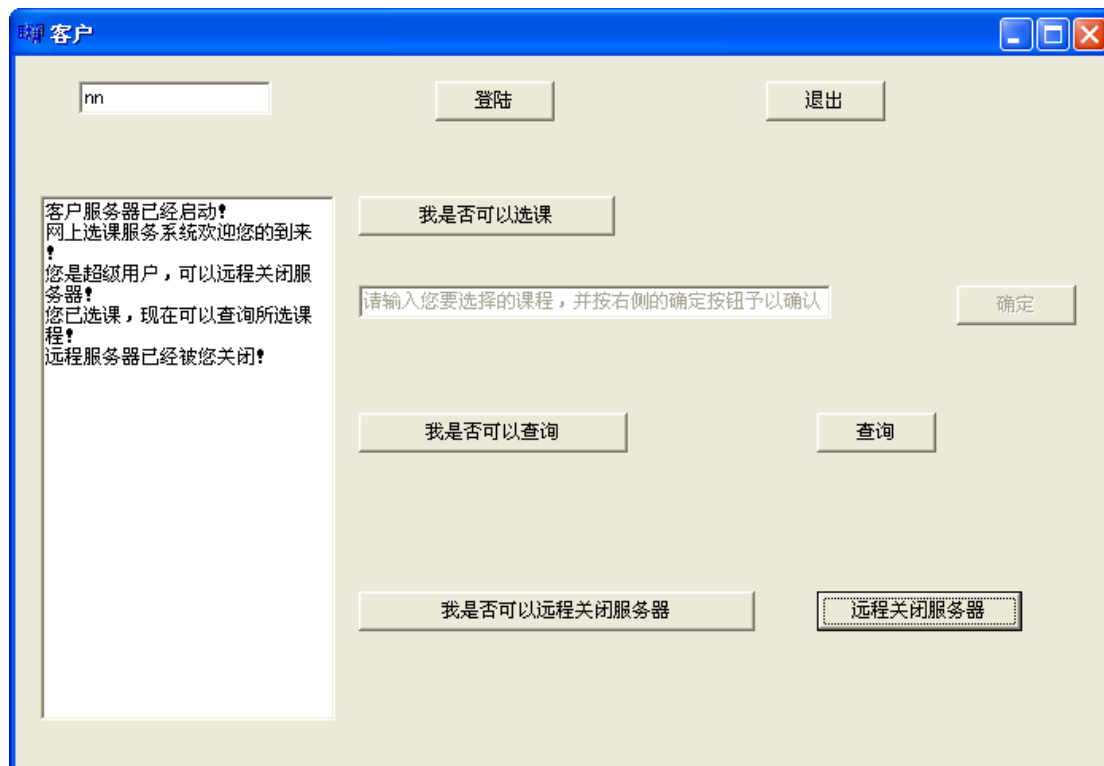


图 12



图 13

11. 一般客户退出服务器以后服务器和客户端的情况分别如图 14 和图 15 所示；

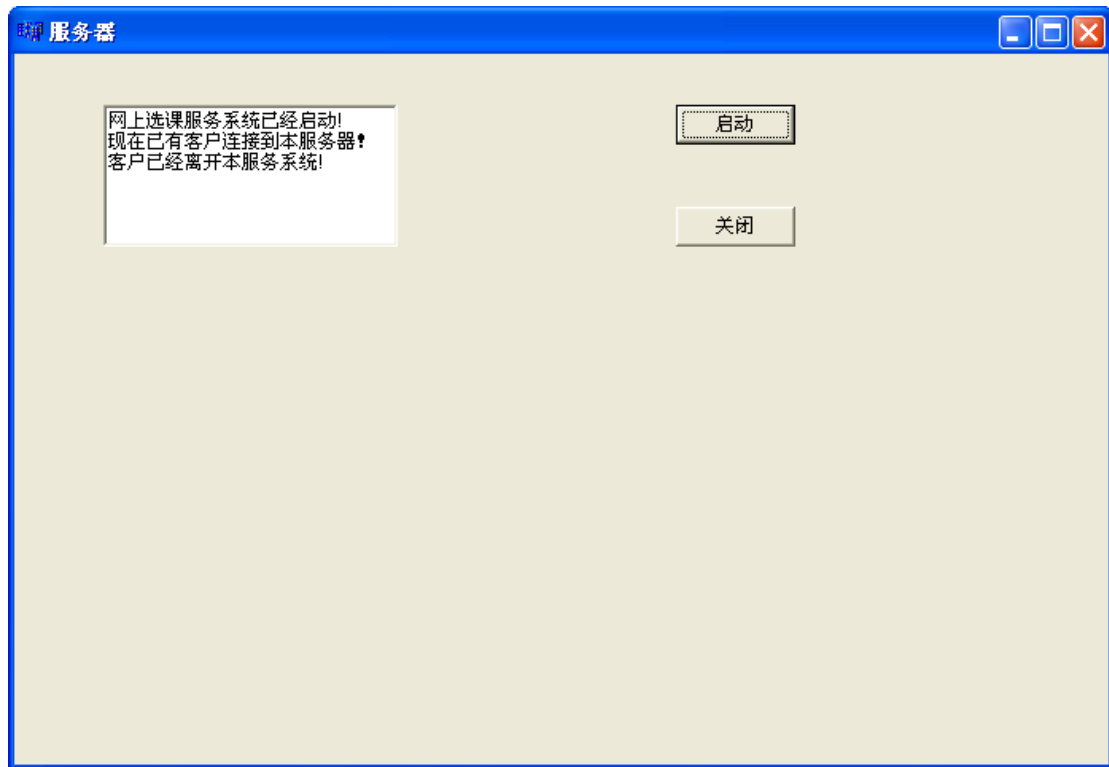


图 14

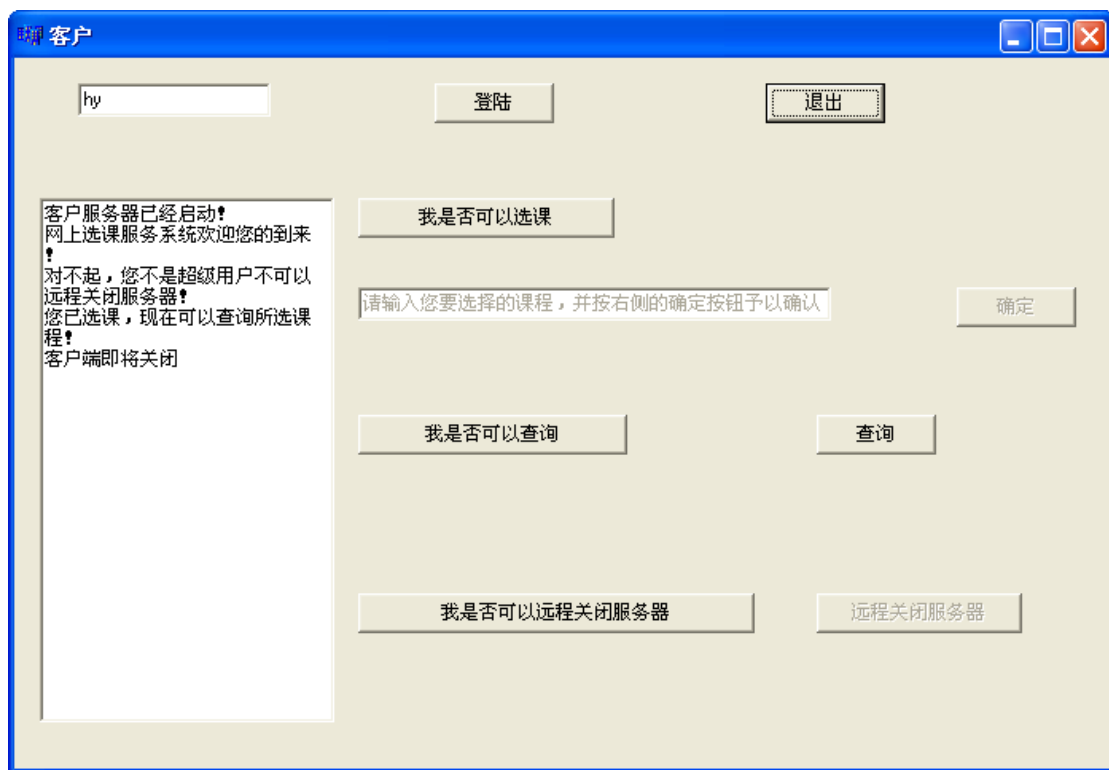


图 15

五、实验故障与排除

1. 在 Client.cpp 中有这样的一个函数

```
void __fastcall TClientForm::LoginButtonClick(TObject *Sender)
{
    //刚开始登陆的时候选课与查询的功能都不可用
    this->ChooseButton->Enabled=false;
    this->FindButton->Enabled=false;

    //以下实现客户登陆到服务器
    ClientSocket->Close();
    ClientSocket->Host="localhost";    //服务器的 IP 地址
    ClientSocket->Port=4700;          //服务器的端口号
    ClientSocket->Tag=0;
    ClientMemo->Lines->Clear();
    ClientSocket->Open();

    i=0;    //客户已与服务器相连但还没有输入用户名
```

可见以下的两句话

```
ClientSocket->Host="localhost";    //服务器的 IP 地址
ClientSocket->Port=4700;          //服务器的端口号
```

设置了客户与服务器进行交互的接口，此处为了简便起见将 IP 设置为了本地主机“localhost”，如果服务器与客户机运行在不同的机器上，则只需要将 ClientSocket->Host 设置为服务器所在机器的 IP 地址，ClientSocket->Port 变为服务器设置的端口号就可以了。

2. 为使得程序运行时不会出错，建议在用户登陆之前一定要先将服务器开启。

六、总结

我利用大约周半的时间编制了这个程序，有一些心得体会在这里和大家分享：在网络编程方面的体会已经在前面的部分详细叙述过了，这里不再赘述。在这里，我只想谈谈在这次完成大作业的过程中，我在学习方法和学习经验方面的一些体会。

(1) 循序渐进

在开始编程的最初一段时间里，由于有些急于求成，很多时候，刚刚看了几眼书，就迫不及待的开始制作，结果看似上手很快，实际事倍功半，基本上没有

做出什么。由于毫无成绩，后来不得不静下心来，一步一步的进行：先了解 C++ Builder 网络编程和数据库编程的基本知识，如 Socket 组件的基本用法、如何制作数据库、添加数据源等，再了解 NCSP 协议的内容，由于我打算采用可视化界面进行编程，所以还需要考虑如何将 NCSP 协议的语言变得更符合普通人的说话习惯。在这个过程中，我使自己能由浅入深，由简到繁地掌握 Socket 编程的基本知识和技术。

(2) 学以致用

从小学到大学，我以前的学习都只是单纯的记忆概念公式，证明、计算习题。即便到了大学有一些实践性的题目（例如：一些编程的题目，实习的题目），也都与实际生活相去甚。在学习编制这个程序的过程中，我并没有把主要精力花费在各个模块孤立地编制和学习上；可以说把学以致用原则贯穿到了整个学习开发的全过程，以使自己对网络的基础知识有更加深刻和形象的理解。

虽然，这个网站与真正的商业化软件有很大的差距，但它对于我的帮助是很大的，可以说，完成这个大作业的过程，不但是我学习巩固知识的过程，而且更是我学习如何学习，学习如何将所学的知识应用于实践的过程。

(3) 养成良好的学习习惯

在开发程序的过程中，最初我图省事，随便将组件命名，（虽然以前上很多课程的时候老师都已经提醒过我们这是很不好的习惯，但是，因为接触的问题规模较小，没有遇到什么麻烦，所以总是不以为意）。但是，这次这个不好的习惯使我吃了不少的苦头。最后，我不得不将所有的组件重新命名。从这一点上，我体会到，在平时的学习实践的过程中，要特别注意纠正自己的坏习惯，养成良好的习惯，将会使你受益匪浅。

附录

程序源码

1. ServerUnit.h

```
#define ServerUnitH
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ScktComp.hpp>
#include <DB.hpp>
#include <DBTables.hpp>
#include <dstring.h>
//-----
class TServerForm : public TForm
{
__published:    // IDE-managed Components
    TMemo *ServerMemo;
    TButton *StartButton;
    TButton *CloseButton;
```

```

TServerSocket *ServerSocket;
TTable *Table1;
TDataSource *DataSource1;
TLabel *NameLabel;
TLabel *Label1;
void __fastcall StartButtonClick(TObject *Sender);
void __fastcall CloseButtonClick(TObject *Sender);
void __fastcall ServerSocketClientDisconnect(TObject *Sender,
    TCustomWinSocket *Socket);
void __fastcall ServerSocketClientRead(TObject *Sender,
    TCustomWinSocket *Socket);
void __fastcall ServerSocketClientConnect(TObject *Sender,
    TCustomWinSocket *Socket);
private:    // User declarations
public:    // User declarations
    int write, check, i;                //write, check 这两个变量用于控制是选课
    //还是查询课程
    //i 表示服务器所处的状态
    __fastcall TServerForm(TComponent* Owner);
};
//-----
extern PACKAGE TServerForm *ServerForm;
//-----
#endif

```

2. ServerUnit.cpp

```

//-----

#include <vcl.h>
#pragma hdrstop

#include "ServerUnit.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TServerForm *ServerForm;
//-----
__fastcall TServerForm::TServerForm(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

```

```
void __fastcall TServerForm::StartButtonClick(TObject *Sender)
{
    //启动服务器
    ServerSocket->Close();
    ServerSocket->Port=4700;
    ServerMemo->Lines->Clear();
    ServerSocket->Open();
    ServerMemo->Lines->Append("网上选课服务系统已经启动!");

    i=0;           //i=0 表示服务器刚刚启动还没有用户登陆
}
//-----
void __fastcall TServerForm::CloseButtonClick(TObject *Sender)
{ //关闭服务器
    ServerSocket->Close();
    ServerMemo->Lines->Append("服务已关闭!");
    i=-1;         //i=-1 表示服务器已关闭
}
//-----

void __fastcall TServerForm::ServerSocketClientDisconnect(TObject *Sender,
    TCustomWinSocket *Socket)
{
    ServerMemo->Lines->Append("客户已经离开本服务系统!");
    i=0;
}
//-----

void __fastcall TServerForm::ServerSocketClientRead(TObject *Sender,
    TCustomWinSocket *Socket)
{
    if(i==1)
    {
        this->NameLabel->Caption=Socket->ReceiveText(); //读入用户名

        if(this->NameLabel->Caption=="") //检查是否输入用户名
        {
            Socket->SendText("尚未输入用户名! 请输入用户名重新登陆!");
            return;
        }
        else //i=2 表示已输入合法的用户名,
            但还未进行具体的选课查询关闭工作
            i=2;
    }
}
```

```
this->Table1->Close(); //判断用户的权限
this->Table1->Open();
this->Table1->IndexFieldNames="NAME";
if(this->Table1->FindKey(new TVarRec(this->NameLabel->Caption),0))
{
    Socket->SendText("canfind");
    check=1;
}
else
{
    Socket->SendText("canchoose");
    write=1;
}
return;
}
if(i==2)
{

this->Label1->Caption=this->ServerSocket->Socket->Connections[0]->ReceiveText()
;
    if(this->Label1->Caption=="onlynncancel") //超级用户远程关闭服务器
    {
        ServerMemo->Lines->Append("远程客户端的超级客户，已经将服务器关闭!");
        this->ServerSocket->Close();
        return;
    }
    if(write==1) //在可以添加数据项的时候进行添加
    {
        if(this->Label1->Caption=="") //检查是否输入所选的课程
        {
            Socket->SendText("尚未输入要选的课程!");
            return;
        }
        try //添加数据项
        {
            this->Table1->Insert();
            this->Table1->FieldByName("NAME")->Value=this->NameLabel->Caption;
            this->Table1->FieldByName("COURSE")->Value=this->Label1->Caption;
            this->Table1->Post();
        }
        catch(...) //出错的处理
        {
```



```

        MessageBox(NULL, "程序出现错误!", "错误信息", MB_OK);
    }
}

if(check==1) //查询课程
{
    this->Table1->Close();
    this->Table1->Open();
    this->Table1->IndexFieldNames="NAME";
    if(!this->Table1->FindKey(new TVarRec(this->NameLabel->Caption), 0))
        Socket->SendText("未在现有数据库找到以"+this->NameLabel->Caption+"为索引的
数据项");
    else
    {
        Socket->SendText("用户"+Table1->FieldByName("NAME")->Value+"选择的课程为
"+Table1->FieldByName("COURSE")->Value);
    }
}
}
}
//-----

void __fastcall TServerForm::ServerSocketClientConnect(TObject *Sender,
    TCustomWinSocket *Socket)
{
    write=check=0; //处置为 0 表示既不可选课又不
    可查讯
    i=1; //i=1 表示用户已连接到服务器
    但还为输入用户名

    ServerMemo->Lines->Append("现在已有客户连接到本服务器!");
    Socket->SendText("网上选课服务系统欢迎您的到来!");
}
//-----

```

3. ClientUnit.h

```

//-----

#ifndef ClientUnitH
#define ClientUnitH
//-----
#include <Classes.hpp>
#include <Controls.hpp>

```

```
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ScktComp.hpp>
//-----
class TClientForm : public TForm
{
__published:    // IDE-managed Components
    TEdit *LoginNameEdit;
    TButton *LoginButton;
    TButton *QuitButton;
    TMemo *ClientMemo;
    TEdit *ChooseEdit;
    TButton *ChooseButton;
    TButton *FindButton;
    TButton *TelCloseButton;
    TClientSocket *ClientSocket;
    TButton *ChooseCheckButton;
    TButton *FindCheckButton;
    TButton *CloseCheckButton;
    TLabel *Label2;
    void __fastcall LoginButtonClick(TObject *Sender);
    void __fastcall QuitButtonClick(TObject *Sender);
    void __fastcall ClientSocketRead(TObject *Sender,
        TCustomWinSocket *Socket);
    void __fastcall ChooseButtonClick(TObject *Sender);
    void __fastcall TelCloseButtonClick(TObject *Sender);
    void __fastcall ClientSocketConnecting(TObject *Sender,
        TCustomWinSocket *Socket);
    void __fastcall FindButtonClick(TObject *Sender);
    void __fastcall FindCheckButtonClick(TObject *Sender);
    void __fastcall ChooseCheckButtonClick(TObject *Sender);
    void __fastcall CloseCheckButtonClick(TObject *Sender);
private:    // User declarations
public:    // User declarations
    int i;    //i 表示客户端处于最初状态，还未与服务器相连
    __fastcall TClientForm(TComponent* Owner);
};
//-----
extern PACKAGE TClientForm *ClientForm;
//-----
#endif
```

4. ClientUnit.cpp

```
//-----  
  
#include <vcl.h>  
#pragma hdrstop  
  
#include "ClientUnit.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TClientForm *ClientForm;  
//-----  
__fastcall TClientForm::TClientForm(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
void __fastcall TClientForm::LoginButtonClick(TObject *Sender)  
{  
  
    //刚开始登陆的时候选课与查询的功能都不可用  
    this->ChooseButton->Enabled=false;  
    this->FindButton->Enabled=false;  
  
    //以下实现客户登陆到服务器  
    ClientSocket->Close();  
    ClientSocket->Host="localhost";  
    ClientSocket->Port=4700;  
    ClientSocket->Tag=0;  
    ClientMemo->Lines->Clear();  
    ClientSocket->Open();  
  
    i=0;    //客户已与服务器相连但还没有输入用户名  
  
}  
//-----  
void __fastcall TClientForm::QuitButtonClick(TObject *Sender)  
{  
    //本函数实现功能将客户端关闭  
    ClientMemo->Lines->Append("客户端即将关闭");  
    ClientSocket->Close();  
}  
//-----
```

```
void __fastcall TClientForm::ClientSocketRead(TObject *Sender,
      TCustomWinSocket *Socket)
{
if(i==0)
    this->ClientMemo->Lines->Append(this->ClientSocket->Socket->ReceiveText());

if(i==1)    //根据服务器的反馈信息决定激活的按钮,也就是说根据服务器的反馈
信息给予用户权限
{
    this->Label2->Caption=this->ClientSocket->Socket->ReceiveText();    //获得
服务器的反馈信息

    if(this->Label2->Caption=="canchoose")
    {
        this->ChooseButton->Enabled=true;
        this->ChooseEdit->Enabled=true;
        ClientMemo->Lines->Append("您现在可以选课!");
    }
    if(this->Label2->Caption=="canfind")
    {
        this->FindButton->Enabled=true;
        ClientMemo->Lines->Append("您已选课, 现在可以查询所选课程!");
    }
    if(this->Label2->Caption=="cantelclose")
    {
        this->TelCloseButton->Enabled=true;
        ClientMemo->Lines->Append("您是超级用户, 可以远程关闭服务器!");
    }

    i=2;    //i=2 表示已经给用户适当的权限, 可以选课, 查询或远程关闭服务器
    return;
}

if(i==2)
    ClientMemo->Lines->Append(Socket->ReceiveText());
    //Socket->SendText("C 端: 来信收到!");
}
//-----

void __fastcall TClientForm::ChooseButtonClick(TObject *Sender)
{
    ClientSocket->Socket->SendText(ChooseEdit->Text);
}
```

```
}
//-----

void __fastcall TClientForm::TelCloseButtonClick(TObject *Sender)
{

this->ClientSocket->Socket->SendText("only"+this->LoginNameEdit->Text+"canclose
");

    this->ClientMemo->Lines->Append("远程服务器已经被您关闭!");

}
//-----

void __fastcall TClientForm::ClientSocketConnecting(TObject *Sender,
    TCustomWinSocket *Socket)
{
    ClientMemo->Lines->Append("客户服务器已经启动!");
    Socket->SendText(this->LoginNameEdit->Text);

}
//-----

void __fastcall TClientForm::FindButtonClick(TObject *Sender)
{
    if(this->ChooseEdit->Text=="")
    {
        ShowMessage("尚未输入要选择的课程，请输入以后再提交!");
        return;
    }
    else
    {
        this->ClientSocket->Socket->SendText(this->ChooseEdit->Text);
    }
}
//-----

void __fastcall TClientForm::FindCheckButtonClick(TObject *Sender)
{
    i=1; //用户将名字告诉了服务器等待服务器告知他有何权限

    this->ClientSocket->Socket->SendText(this->LoginNameEdit->Text);
```

```
}  
//-----  
  
void __fastcall TClientForm::ChooseCheckButtonClick(TObject *Sender)  
{  
    i=1; //用户将名字告诉了服务器等待服务器告知他有何权限  
  
    this->ClientSocket->Socket->SendText(this->LoginNameEdit->Text);  
}  
//-----  
  
void __fastcall TClientForm::CloseCheckButtonClick(TObject *Sender)  
{  
    i=1; //用户将名字告诉了服务器等待服务器告知他有何权限  
    if(this->LoginNameEdit->Text=="nn")  
    {  
        this->FindButton->Enabled=false;  
        this->TelCloseButton->Enabled=true;  
        ClientMemo->Lines->Append("您是超级用户，可以远程关闭服务器！");  
    }  
    else  
        ClientMemo->Lines->Append("对不起，您不是超级用户不可以远程关闭服务器！");  
  
    this->ClientSocket->Socket->SendText(this->LoginNameEdit->Text);  
}  
//-----
```