

# 西安交通大学实验报告

课程 网络 实验名称 基于 C/S 模式的选课系统

系 别 电气工程及其自动化 实 验 日 期 2004 年 5 月 16 日

专业班级 电气 16 组别 \_\_\_\_\_ 实 验 报 告 日 期 2004 年 5 月 26 日

姓 名 康鑫 学号 01041166 报 告 退 发 ( 订 正 、 重 做 )

同 组 人 \_\_\_\_\_ 教 师 审 批 签 字

---

## 实验名称 基于 C/S 模式的选课系统

### 一、 题目

使用任意网络编程语言 (Java、C、VB、Delphi 等) 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 TCP 协议, 并且实现 NCSP 应用层协议 (Network-based Course Selection Protocol, 见第四节)。

### 二、 环境

运行环境 : Windows9x/NT/2000/ XP

开发环境 : Windows Xp

VC + + 6.0

程序简介:

**编译:** 在开发环境 VC + + 6.0 中分别对 client 和 server 进行编译

**运行:** 程序在编译后, 会被自动封装成 exe 文件放在 debug 中,

在 Dos 命令符下直接运行 client.exe 和 server.exe 文件即可。

具体的运行顺序是: 先运行服务器端程序, 然后运行客户端程序。

### 三、逻辑框图(程序)

程序编写流程框图如下:

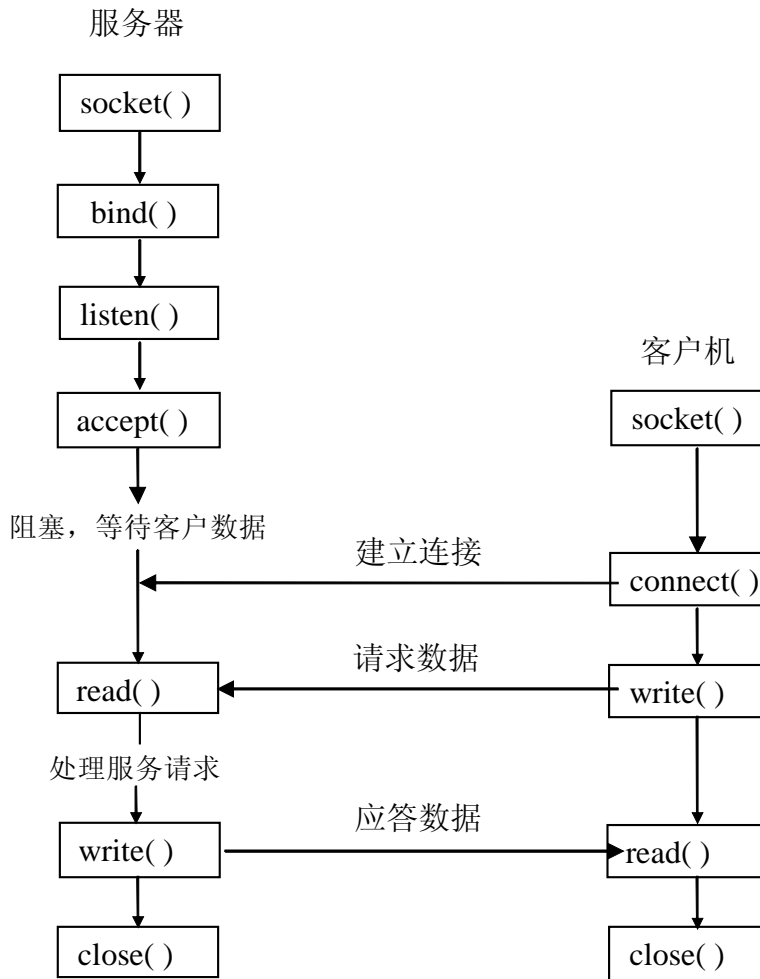


图 1 客户端/服务器之间的信息交互流程图

### 四、设计思想

大多数网络应用程序都遵从 client/server 交互模型。该模型中，服务器应用是一种专门用来提供某一种服务的程序，它被动地等待通信，而客户应用主动地启动通信，请求服务器的服务。服务器程序既可以接收信息也能发送信息，还可以成为客户。

TCP/UDP 赋予每个服务一个唯一的协议端口号。服务器程序通过协议端口号来指定它所提供的服务，然后被动地等待通信。客户在发送连接请求时，必须说明服务器程序运行主机的 IP 地址和协议端口号来指定它所希望的服务。服务器

端计算机通过此端口号将收到的请求转向正确的服务器程序。

SETCOURSE 指令:

当接收到 client 发送的 GETCOURSE 协议消息, 返回该用户已经选择的所有课程名字;

GETCOURSE 指令:

当接收到 client 发送的 SETCOURSE 协议消息, 记录用户名和课程名, 以便 client 使用 GETCOURSE 命令查询;

SHUTDOWN 指令:

当接收到 client 发送的 SHUTDOWN 协议消息, 检查用户是否有此权限, 如果有则结束 server 程序。

## 五、关键算法分析

该程序主要利用字符串库函数, 对输入命令进行判断和进行相应的操作并且定义学生结构体, 用来存贮学生的选课, 名字和权限。

并且设置了很多子函数, 分别来实现各自的功能, 使程序简单易懂,

主要设置函数和变量如下

```
struct students{char corse[10][10];char name[10];char priv;}student[20];
```

```
//定义学生结构体, 用来存贮学生的选课, 名字和权限。
```

```
char szBuf[256];//定义的缓存区
```

```
char order[9];//用来存贮从客户端接收的命令
```

```
char name[10];//用来存贮从客户端接收的选课学生名
```

```
void initial();//初始化函数,
```

```
void send();//发送函数
```

```
void receive();//接收函数
```

```
void handle();//核心处理函数
```

```
void getcorse();//处理 getcorse 命令的函数
```

```
void setcorse();//处理 setcorse 命令的函数
```

```
void clear();//对缓存区清零的函数
```

## 六、程序源代码

### 客户端:

```
#include <stdio.h>
#include<iostream.h>
#include <winsock.h>

////////////////////////////////////
char szBuf[256];//定义缓冲区
SOCKET theSocket;
////////////////////////////////////
// Function prototype
void StreamClient(char *szServer, short nPort);

////////////////////////////////////增设函数////////////////////////////////////
void send();//用来向 server 发送信息的发送函数
void receive();//用来从 server 接收信息的接收函数
void handle();//核心处理函数
////////////////////////////////////

// Helper macro for displaying errors
#define PRINTEROR(s) \
    fprintf(stderr, "\n%: %d\n", s, WSAGetLastError())

////////////////////////////////////

void main(int argc, char **argv)
{
    WORD wVersionRequested = MAKEWORD(1,1);
    WSADATA wsaData;
    int nRet;
    short nPort;

    //
    // Check for the host and port arguments
    //
    if (argc != 3)
    {
        fprintf(stderr, "\nSyntax: client ServerName PortNumber\n");
        return;
    }

    nPort = atoi(argv[2]);
```

```

//
// Initialize WinSock and check the version
//
nRet = WSASStartup(wVersionRequested, &wsaData);
if (wsaData.wVersion != wVersionRequested)
{
    fprintf(stderr, "\n Wrong version\n");
    return;
}

//
// Go do the stuff a stream client does
//
puts("Client is running now!");//与 server 连接后显示提示信息
StreamClient(argv[1], nPort);
handle();//进入核心处理程序

//
// Release WinSock
//
WSACleanup();
}

////////////////////////////////////

void StreamClient(char *szServer, short nPort)
{
    printf("\nStream Client connecting to server: %s on port: %d",
        szServer, nPort);

    //
    // Find the server
    //
    LPHOSTENT lpHostEntry;

    lpHostEntry = gethostbyname(szServer);
    if (lpHostEntry == NULL)
    {
        PRINTERROR("gethostbyname()");
        return;
    }
}

```

```

//
// Create a TCP/IP stream socket
//

theSocket = socket(AF_INET,           // Address family
                  SOCK_STREAM,       // Socket type
                  IPPROTO_TCP);     // Protocol
if (theSocket == INVALID_SOCKET)
{
    PRINTERROR("socket()");
    return;
}

//
// Fill in the address structure
//
SOCKADDR_IN saServer;

saServer.sin_family = AF_INET;
saServer.sin_addr = *((LPIN_ADDR)*IpHostEntry->h_addr_list);
// ^ Server's address
saServer.sin_port = htons(nPort); // Port number from command line

//
// connect to the server
//
int nRet;

nRet = connect(theSocket,           // Socket
              (LPSOCKADDR)&saServer, // Server address
              sizeof(struct sockaddr)); // Length of server address structure
if (nRet == SOCKET_ERROR)
{
    PRINTERROR("socket()");
    closesocket(theSocket);
    return;
}

//
// Send data to the server
//

```

```
}
```

void receive()//用来从 server 接收信息的接收函数

```
{
    int nRet;
    ///////////////////////////////////
    int i;
    for(i=0;i<256;i++)
    {   szBuf[i]=NULL;}
    ///////////////////////////////////

    nRet = recv(theSocket,           // Connected socket
                szBuf,              // Receive buffer
                sizeof(szBuf),      // Size of receive buffer
                0);                 // Flags
    if (nRet == SOCKET_ERROR)
    {
        PRINTERROR("recv()");
        closesocket(theSocket);
        return;
    }

    //
    // Display the received data
    //
    puts(szBuf);
}
```

void send()//用来向 server 发送信息的发送函数

```
{
    int nRet;
    nRet = send(theSocket,          // Connected socket
                szBuf,              // Data buffer
                strlen(szBuf),      // Length of data
                0);                 // Flags
    if (nRet == SOCKET_ERROR)
    {
        PRINTERROR("send()");
        closesocket(theSocket);
        return;
    }
}
```

```

}
void handle()//核心处理函数
{
    for(;;)
    {
        //建立连接后，显示操作提示信息，提示用户进行相应的操作//
        cout<<endl;
        cout<<"Please input the order in the following ways:"<<endl;
        cout<<"1.TO see what you have chosen ,please input
GETCORSE+space+username;"<<endl;
        cout<<"2.TO choose your corse , please input
SETCORSE+space+username;"<<endl;
        cout<<"3.TO shutdown the application ,please input
SHUTDOWN+space+username;"<<endl;
        cout<<endl;
        //将输入的命令存入缓存区
        gets(szBuf);
        if(strstr(szBuf,"GETCORSE")!=NULL)//从缓存区中查找 GETCORSE,如果发现是该
命令则从 server 端接收信息
        {
            send();
            receive();
            if(strcmp(szBuf,"401 User does not exist")==0){continue;}//如果用户不存在，则
返回主菜单（即显示操作提示信息）
            if(strcmp(szBuf,"200 OK")==0)//如果接收到的是 200 OK，则从 server 端接收信
息
            {
                while(strcmp(szBuf,".")!=0//直到接收到".", 然后返回主菜单（即显示操作
提示信息）
                {
                    receive();
                }
                continue;
            }
        }
        else if(strstr(szBuf,"SETCORSE")!=NULL)//从缓存区中查找 SETCORSE, 如果发现
是该命令则向 server 端发送信息
        {
            send();
            receive();
            if(strstr(szBuf,"301 User exists")!=NULL)continue;//如果用户存在，则返回主菜
单（即显示操作提示信息）
            if(strstr(szBuf,"200 OK")!=NULL)//如果接收到的是 200 OK，则将从缓冲区获
得的选课信息，发送给服务器

```



```

        {
            do
            {
                gets(szBuf);
                send();
            }while(strcmp(szBuf, ".")!=0);
            continue;
        }
    }
    else if(strstr(szBuf, "SHUTDOWN")!=NULL)//从缓存区中查找 SHUTDOWN, 如果发
    现是该命令则向 server 端发送 SHUTDOWN 信息
    {
        send();
        receive();
        if(strcmp(szBuf, "200 OK")==0)//如果接收到的是 200 OK, 则关闭端后, 返回主
        程序, 即退出
        {
            printf("SHUT DOWN!\n");
            closesocket(theSocket);
            return;
        }
        else continue;//否则返回主菜单 (即显示操作提示信息)
    }
    else //如没有以上三个命令, 否则返回主菜单 (即显示操作提示信息)
    {
        send();
        receive();
        continue;
    }
}
}
}

```

## 服务器端:

```

#include <stdio.h>
#include <winsock.h>
////////////////////////////////////
struct students{ char corse[10][10];char name[10];char priv;}student[20];
//定义学生结构体, 用来存贮学生的选课, 名字和权限。
char szBuf[256];//定义的缓存区
SOCKET remoteSocket;
SOCKET listenSocket;
int nRet;

```

```

char order[9];//用来存贮从客户端接收的命令
char name[10];//用来存贮从客户端接收的选课学生名

////////////////////////////////////
// Function prototype
void StreamServer(short nPort);
////////////////////////////////////
void initial();//初始化函数,
void send();//用来向客户端发送信息的发送函数
void receive();//用来从客户端接收信息的接收函数
void handle();//核心处理函数
void getcourse();//处理 getcourse 命令的函数
void setcourse();//处理 setcourse 命令的函数
void clear();//对缓存区清零的函数
////////////////////////////////////

// Helper macro for displaying errors
#define PRINTERERROR(s) \
    fprintf(stderr, "\n%: %d\n", s, WSAGetLastError())

////////////////////////////////////

void main(int argc, char **argv)
{
    WORD wVersionRequested = MAKEWORD(1,1);
    WSADATA wsaData;
    int nRet;
    short nPort;

    //
    // Check for port argument
    //
    if (argc != 2)
    {
        fprintf(stderr, "\nSyntax: server PortNumber\n");
        return;
    }

    nPort = atoi(argv[1]);

    //
    // Initialize WinSock and check version
    //
    nRet = WSASStartup(wVersionRequested, &wsaData);

```

```

if (wsaData.wVersion != wVersionRequested)
{
    fprintf(stderr, "\n Wrong version\n");
    return;
}

//
// Do the stuff a stream server does
//
StreamServer(nPort);

//
// Release WinSock
//
WSACleanup();
}

////////////////////////////////////

void StreamServer(short nPort)
{
    //
    // Create a TCP/IP stream socket to "listen" with
    //
    //////////////////////////////////
// SOCKET listenSocket;
    //////////////////////////////////

    listenSocket = socket(AF_INET,          // Address family
                        SOCK_STREAM,      // Socket type
                        IPPROTO_TCP);    // Protocol

    if (listenSocket == INVALID_SOCKET)
    {
        PRINTERROR("socket()");
        return;
    }

    //
    // Fill in the address structure
    //
    SOCKADDR_IN saServer;

```

```

saServer.sin_family = AF_INET;
saServer.sin_addr.s_addr = INADDR_ANY; // Let WinSock supply address
saServer.sin_port = htons(nPort);      // Use port from command line

//
// bind the name to the socket
//

////////////////////////////////////

nRet = bind(listenSocket,          // Socket
            (LPSOCKADDR)&saServer, // Our address
            sizeof(struct sockaddr)); // Size of address structure
if (nRet == SOCKET_ERROR)
{
    PRINTERROR("bind()");
    closesocket(listenSocket);
    return;
}

//
// This isn't normally done or required, but in this
// example we're printing out where the server is waiting
// so that you can connect the example client.
//
int nLen;
nLen = sizeof(SOCKADDR);
////////////////////////////////////

nRet = gethostname(szBuf, sizeof(szBuf));
if (nRet == SOCKET_ERROR)
{
    PRINTERROR("gethostname()");
    closesocket(listenSocket);
    return;
}

//
// Show the server name and port number
//
printf("\nServer named %s waiting on port %d\n",
       szBuf, nPort);

```

```

//
// Set the socket to listen
//

printf("\nlisten()");
nRet = listen(listenSocket,          // Bound socket
              SOMAXCONN);          // Number of connection request
queue
if (nRet == SOCKET_ERROR)
{
    PRINTERROR("listen()");
    closesocket(listenSocket);
    return;
}

//
// Wait for an incoming request
//

////////////////////////////////////

printf("\nBlocking at accept()");
remoteSocket = accept(listenSocket,  // Listening socket
                      NULL,         // Optional client address
                      NULL);
if (remoteSocket == INVALID_SOCKET)
{
    PRINTERROR("accept()");
    closesocket(listenSocket);
    return;
}

//
// We're connected to a client
// New socket descriptor returned already
// has clients address

//
// Receive data from the client
//
puts("Server is running now!");//与客户端连接后显示提示信息
handle();//进入核心处理程序
//

```

```

// Close BOTH sockets before exiting
//
closesocket(remoteSocket);
closesocket(listenSocket);
return;
}

//////////初始化//////////
void initial()
{
    int i,j,p;
    ////////////设置 kangxin 为特权用户（即该用户可以远程关闭该服务器）//////////
    student[0].priv=1;
    strcpy(student[0].name,"kangxin");
    strcpy(student[0].corse[0],".");
    ////////////结构体清零，进行初始化//////////
    for(p=1;p<20;p++)
    {
        for(i=0;i<10;i++)
        {
            for(j=0;j<10;j++)
            {
                student[p].corse[i][j]=NULL;
            }
            student[p].name[i]=NULL;
        }
        student[p].priv=0;
    }
}

//////////发送//////////
void send()
{
    nRet = send(remoteSocket,                // Connected socket
                szBuf,                       // Data buffer
                strlen(szBuf),              // Lenght of data
                0);                          // Flags
}

//////////接收//////////
void receive()
{
    clear();
}

```

```

memset(szBuf, 0, sizeof(szBuf));
nRet = recv(remoteSocket,          // Connected client
            szBuf,                  // Receive buffer
            sizeof(szBuf),         // Length of buffer
            0);                    // Flags
if (nRet == INVALID_SOCKET)
{
    PRINTERROR("recv()");
    closesocket(listenSocket);
    closesocket(remoteSocket);
    return;
}
puts(szBuf);
}
//////////处理函数//////////
void handle()
{
    initial();//进行初始化
    for(;;)
    {
        int i;
        //////////接收客户端发来的命令，存入 order 中//////////
        receive();
        for(i=0;i<8;i++)
        {
            order[i]=szBuf[i];
            order[8]='\0';
        }
        if(strlen(szBuf)>17){continue;}
        i=-1;
        do
        {
            i++;
            name[i]=szBuf[i+9];
        }while(szBuf[i]!='\0');//将用户名临时存入 name 中
        puts(name);//将用户名显示在 server 端
        if(strcmp(order,"GETCORSE")==0)//如果是 GETCORSE，则进入 getcorse() 函数进
行处理
        {
            getcorse();
            continue;
        }
        else if(strcmp(order,"SETCORSE")==0)//如果是 SETCORSE，则进入 SETCORSE()
函数进行处理

```

```

    {
        setcorse();
        continue;
    }
    //如果是 SHUTDOWN, 则先判断是不是特权用户, 若是特权用户则关闭服务器
    //若不是特权用户, 则显示用户没用权限
else if(strcmp(order,"SHUTDOWN")==0)
{
    for(i=0;i<8;i++)
        order[i]=NULL;
    puts("shutdown");
    for(i=0;i<20;i++)
    {
        if(strcmp(name,student[i].name)==0&&student[i].priv==1)
        {
            puts("shutting down now.....");
            clear();
            strcpy(szBuf,"200 OK");
            send();
            return;
        }
        else
if(student[i].name[0]==NULL||(strcmp(name,student[i].name)==0&&student[i].priv!=1))
        {
            puts("you don't have the privilege!");
            clear();
            strcpy(szBuf,"201 User not allowed to execute this command");
            send();
            break;
        }
    }
    continue;
}
else //如果不是上述三个命令, 则显示命令格式不正确
{
    clear();
    strcpy(szBuf,"300 Message format error");
    send();
}
}
}
//////////GETCORSE 函数//////////
void getcorse()
{

```



```

puts("get is ready");
int i,j;
for(i=0;i<8;i++)
    order[i]=NULL;//将存贮命令的数组清零
//将接收命令后，临时存贮在 name 中的用户名和结构体中的用户名进行比较//
//如果该用户名在结构体中存在，则说明该用户已选课，将其已选课科目发给客户端//
//如果该用户名在结构体中不存在，则说明该用户未选课，则向客户端发送该客户不存在的信息//
for(i=1;i<20;i++)
{
    puts(student[i].name);
    if(strcmp(name,student[i].name)==0)
    {
        puts("GETTING COURSE");
        clear();
        strcpy(szBuf,"200 OK");
        send();
        j=-1;
        do
        {
            clear();
            puts("GET OK");
            j++;
            strcpy(szBuf,student[i].course[j]);
            send();
        }while(strcmp(student[i].course[j],".")!=0);
        return;
    }
    else if(student[i].name[0]==NULL)
    {
        puts("NO THIS USER");
        clear();
        strcpy(szBuf,"401 User does not exist");
        send();
        return;
    }
}
}
/////////////////////////////////SETCORSE 函数/////////////////////////////////
void setcourse()
{
    puts("set is ready");
    int i,j;
    for(i=0;i<8;i++)
        order[i]=NULL;

```

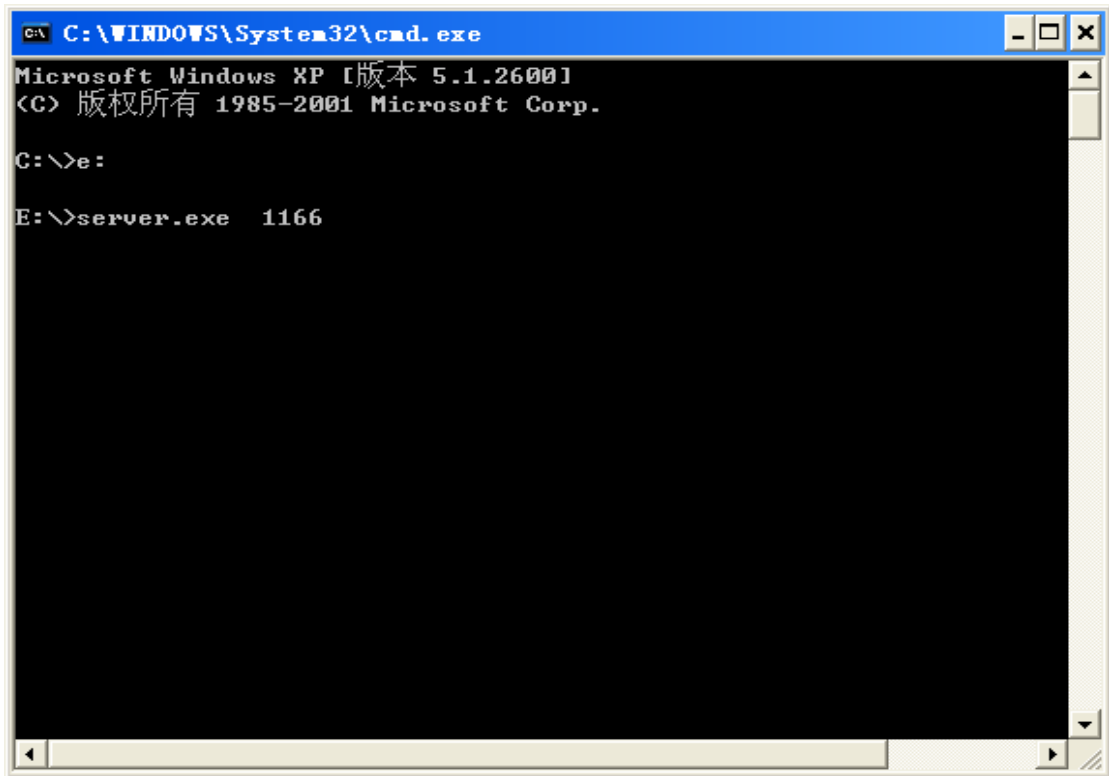
//将接收命令后，临时存贮在 name 中的用户名和结构体中的用户名进行比较//  
//如果该用户名在结构体中存在，则说明该用户已选课，则向客户端发送该用户已存在信息，  
不允许进行选课操作//

//如果该用户名在结构体中不存在，则说明该用户未选课，则向客户端发送 200 OK 信息，  
允许选课，并将用户名和其选课存入结构体中//

```
for(i=1;i<20;i++)
{
    if(strcmp(name,student[i].name)==0)
    {
        puts("CAN'T SET THIS USER");
        clear();
        strcpy(szBuf,"301 User exists");
        send();
        return;
    }
    else if(student[i].name[0]==NULL)
    {
        clear();
        strcpy(szBuf,"200 OK");
        send();
        strcpy(student[i].name,name);
        j=-1;
        do
        {
            clear();
            puts("SET OK");
            j++;
            receive();
            strcpy(student[i].corse[j],szBuf);
        }while(strcmp(szBuf,".")!=0);
        return;
    }
}
}
}
//////////清除缓冲区//////////
void clear()
{
    int i;
    for(i=0;i<256;i++)
    {
        szBuf[i]=NULL;
    }
}
```

## 七、实验过程及演示

### 1、启动服务器：输入可执行程序路径地址和 server 的端口号

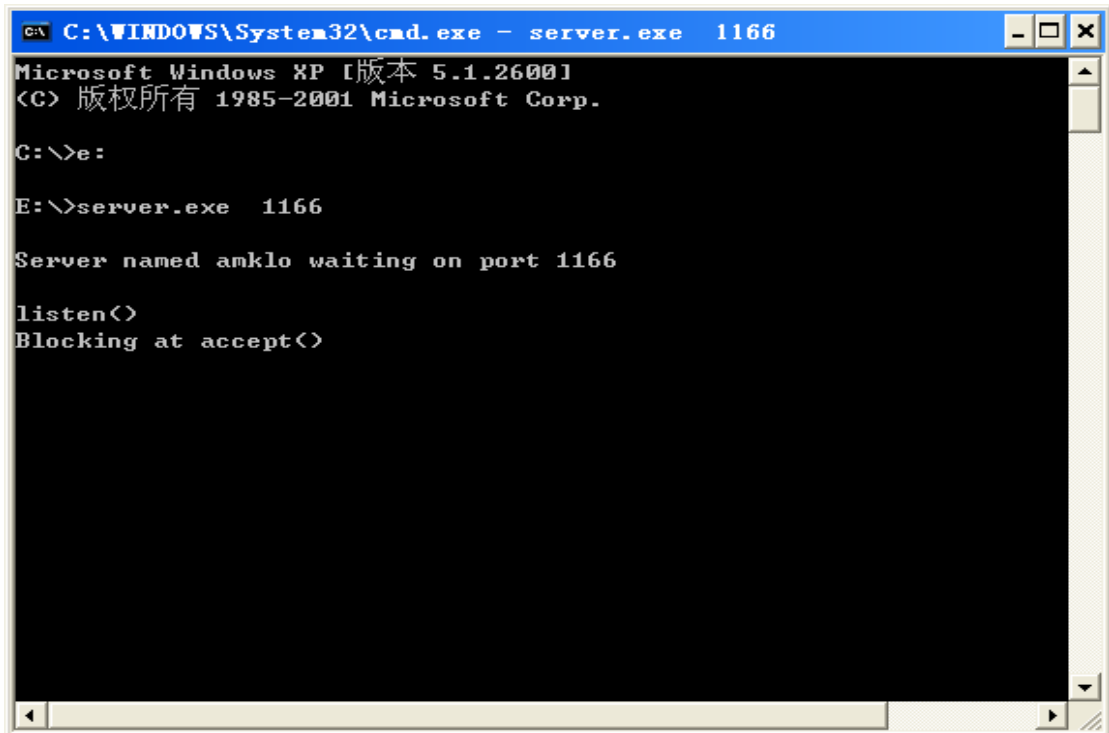


```
C:\WINDOWS\System32\cmd.exe
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\>e:

E:\>server.exe 1166
```

### 2、服务器启动，等待连接



```
C:\WINDOWS\System32\cmd.exe - server.exe 1166
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

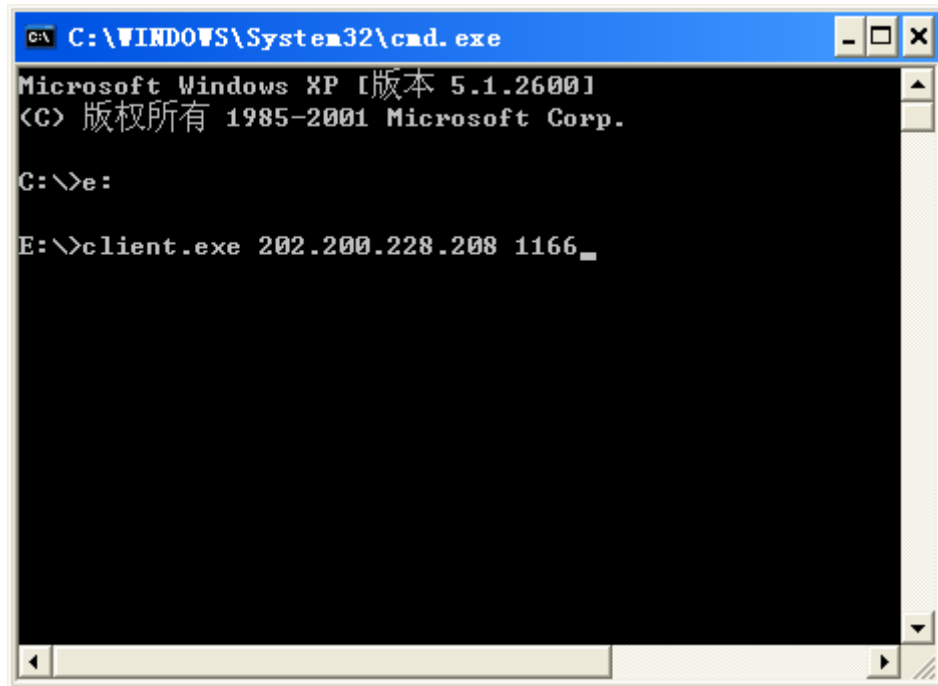
C:\>e:

E:\>server.exe 1166

Server named amklo waiting on port 1166

listen<>
Blocking at accept<>
```

- 3、启动客户端：输入可执行程序路径地址和 server 的 IP 地址和 server 的端口号



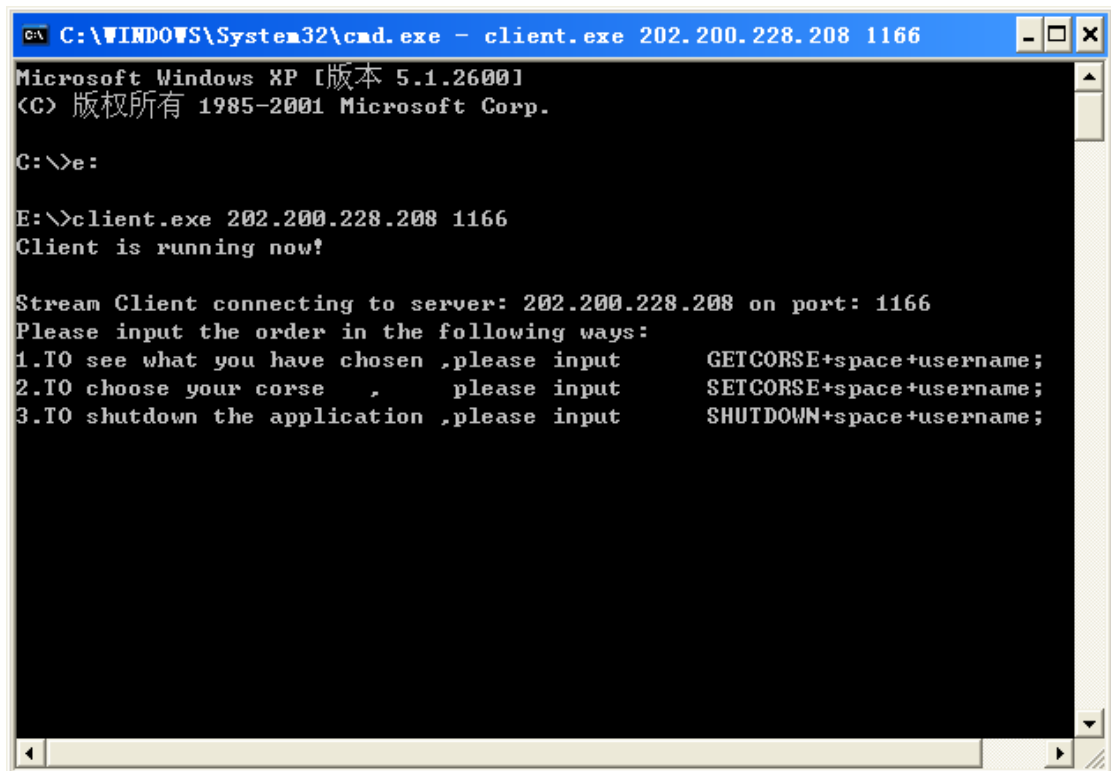
```
C:\WINDOWS\System32\cmd.exe
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\>e:

E:\>client.exe 202.200.228.208 1166_
```

- 4、输入回车，客户端连到服务器

客户端显示如下：



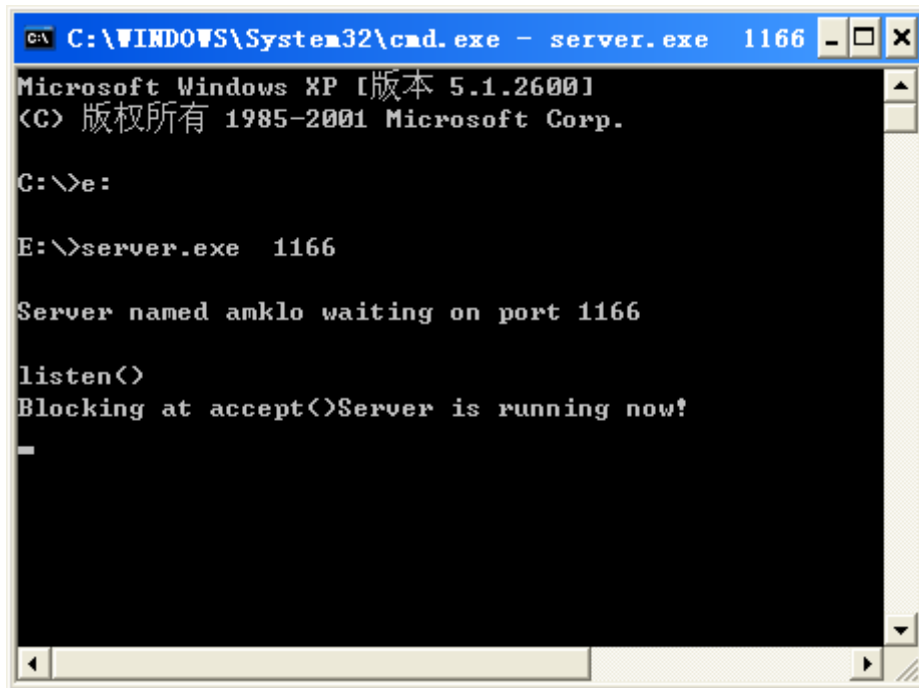
```
C:\WINDOWS\System32\cmd.exe - client.exe 202.200.228.208 1166
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\>e:

E:\>client.exe 202.200.228.208 1166
Client is running now!

Stream Client connecting to server: 202.200.228.208 on port: 1166
Please input the order in the following ways:
1.TO see what you have chosen ,please input      GETCORSE+space+username;
2.TO choose your corse      ,      please input      SETCORSE+space+username;
3.TO shutdown the application ,please input      SHUTDOWN+space+username;
```

服务器端显示如下:



```
C:\WINDOWS\System32\cmd.exe - server.exe 1166
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

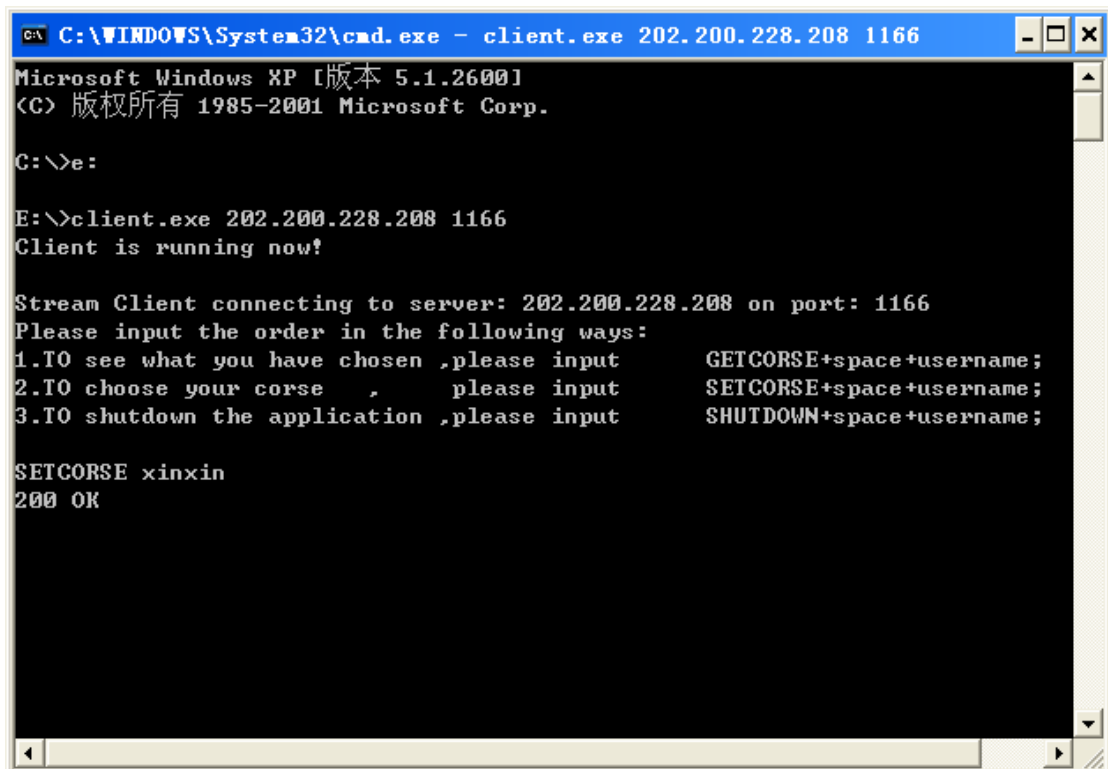
C:\>e:

E:\>server.exe 1166

Server named amklo waiting on port 1166

listen()
Blocking at accept()Server is running now!
```

5、输入 SETCORSE 命令和用户名，用户名如不存在，则显示 200 OK，表示可以选课



```
C:\WINDOWS\System32\cmd.exe - client.exe 202.200.228.208 1166
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

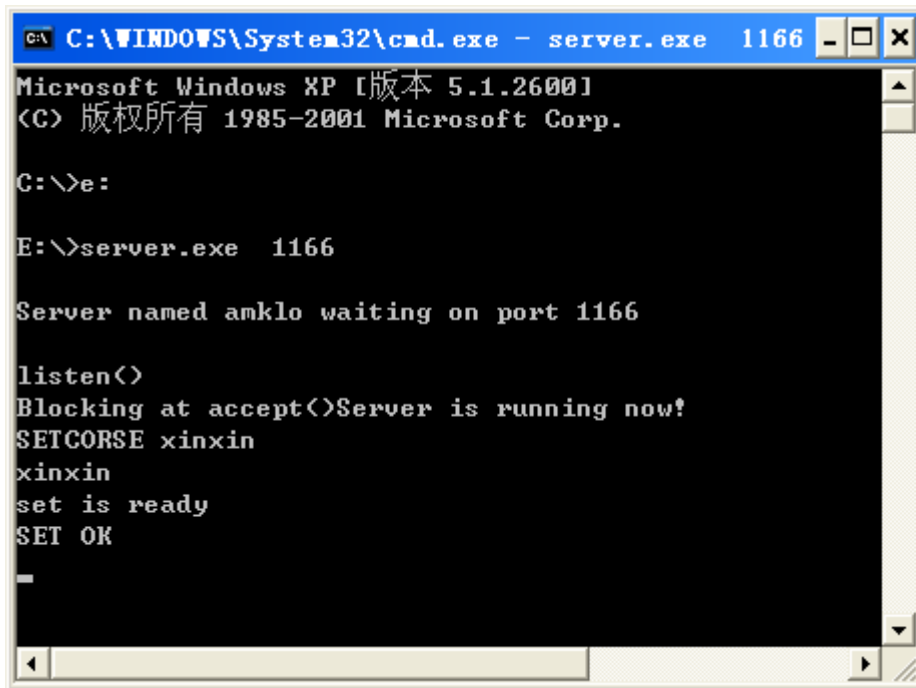
C:\>e:

E:\>client.exe 202.200.228.208 1166
Client is running now!

Stream Client connecting to server: 202.200.228.208 on port: 1166
Please input the order in the following ways:
1.TO see what you have chosen ,please input      GETCORSE+space+username;
2.TO choose your corse ,      please input      SETCORSE+space+username;
3.TO shutdown the application ,please input      SHUTDOWN+space+username;

SETCORSE xinxin
200 OK
```

服务器端显示如下:



```
C:\WINDOWS\System32\cmd.exe - server.exe 1166
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

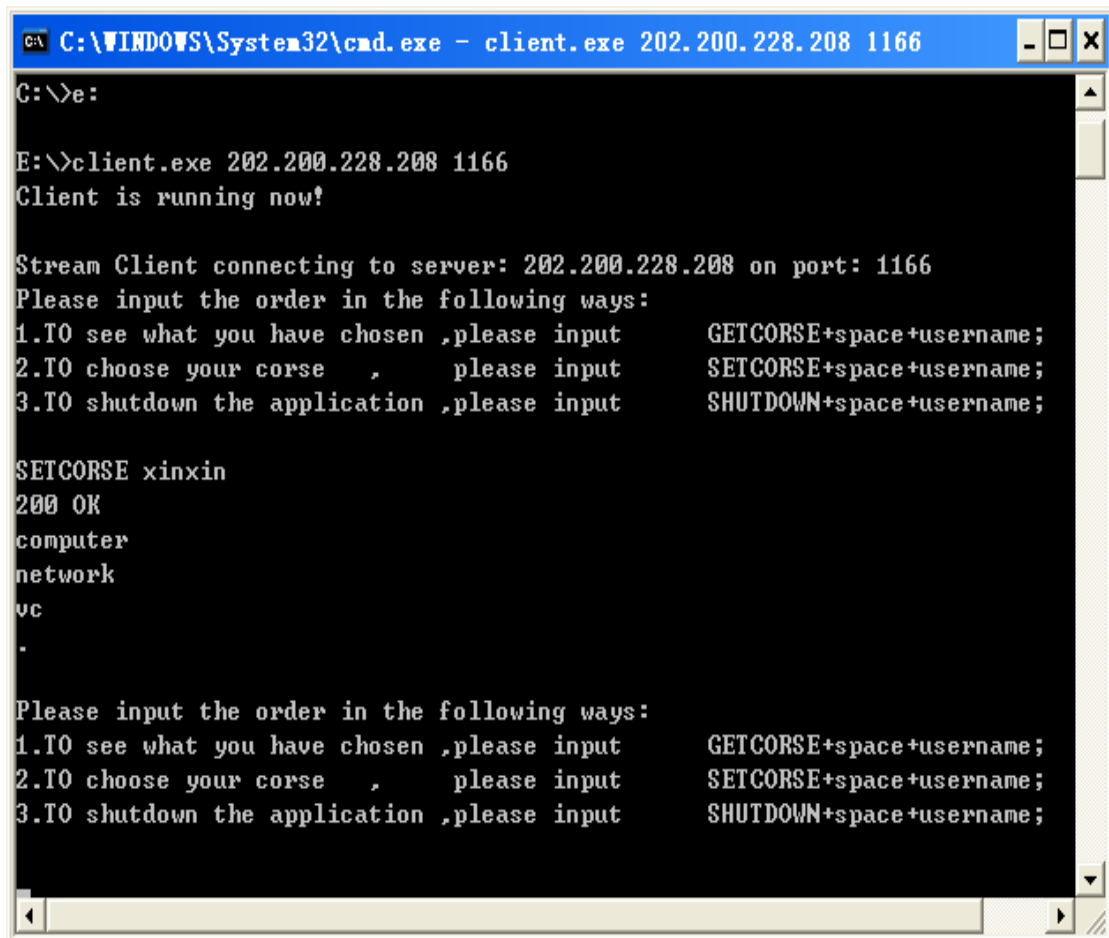
C:\>e:

E:\>server.exe 1166

Server named anklo waiting on port 1166

listen()
Blocking at accept()Server is running now!
SETCORSE xinxin
xinxin
set is ready
SET OK
```

6、输入要选的课程，以“.”结束



```
C:\WINDOWS\System32\cmd.exe - client.exe 202.200.228.208 1166
C:\>e:

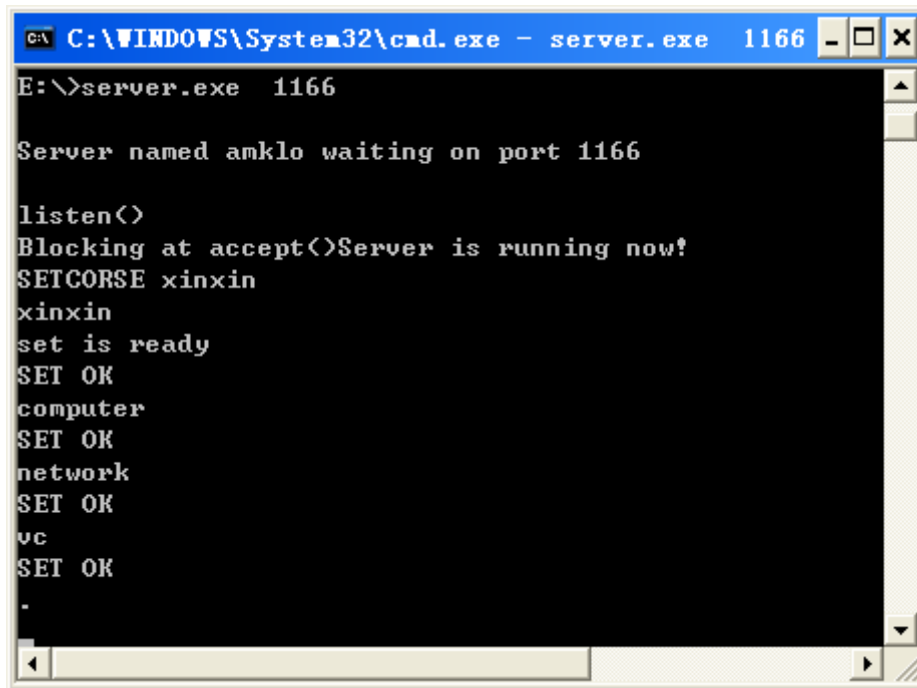
E:\>client.exe 202.200.228.208 1166
Client is running now!

Stream Client connecting to server: 202.200.228.208 on port: 1166
Please input the order in the following ways:
1.TO see what you have chosen ,please input      GETCORSE+space+username;
2.TO choose your corse      ,      please input    SETCORSE+space+username;
3.TO shutdown the application ,please input      SHUTDOWN+space+username;

SETCORSE xinxin
200 OK
computer
network
vc
.

Please input the order in the following ways:
1.TO see what you have chosen ,please input      GETCORSE+space+username;
2.TO choose your corse      ,      please input    SETCORSE+space+username;
3.TO shutdown the application ,please input      SHUTDOWN+space+username;
```

服务器端显示如下:

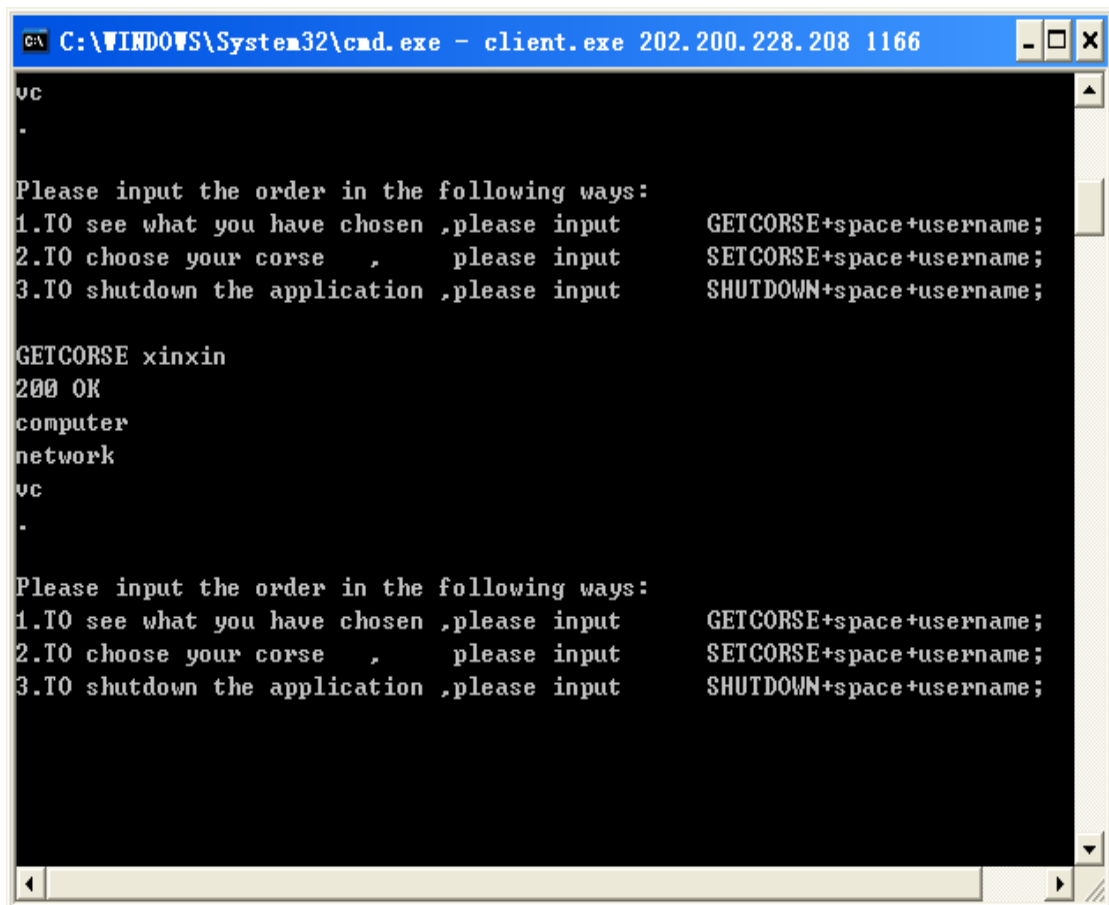


```
C:\WINDOWS\System32\cmd.exe - server.exe 1166
E:\>server.exe 1166

Server named amklo waiting on port 1166

listen()
Blocking at accept()Server is running now!
SETCORSE xinxin
xinxin
set is ready
SET OK
computer
SET OK
network
SET OK
vc
SET OK
.
```

7、输入 GETCORSE 命令和用户名，可以显示该用户所选的课程



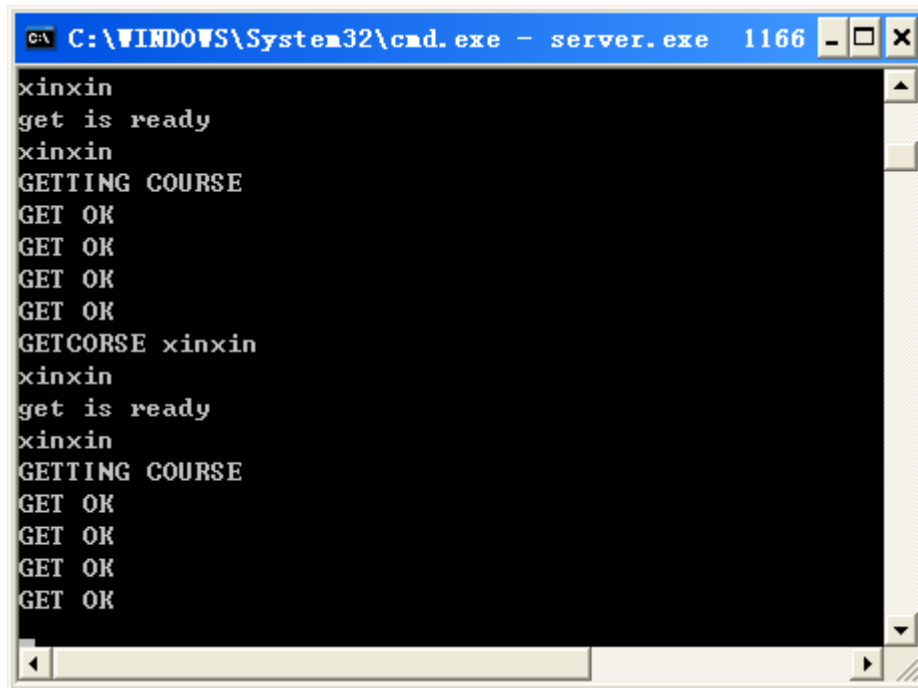
```
C:\WINDOWS\System32\cmd.exe - client.exe 202.200.228.208 1166
vc
.

Please input the order in the following ways:
1.TO see what you have chosen ,please input      GETCORSE+space+username;
2.TO choose your corse ,      please input      SETCORSE+space+username;
3.TO shutdown the application ,please input      SHUTDOWN+space+username;

GETCORSE xinxin
200 OK
computer
network
vc
.

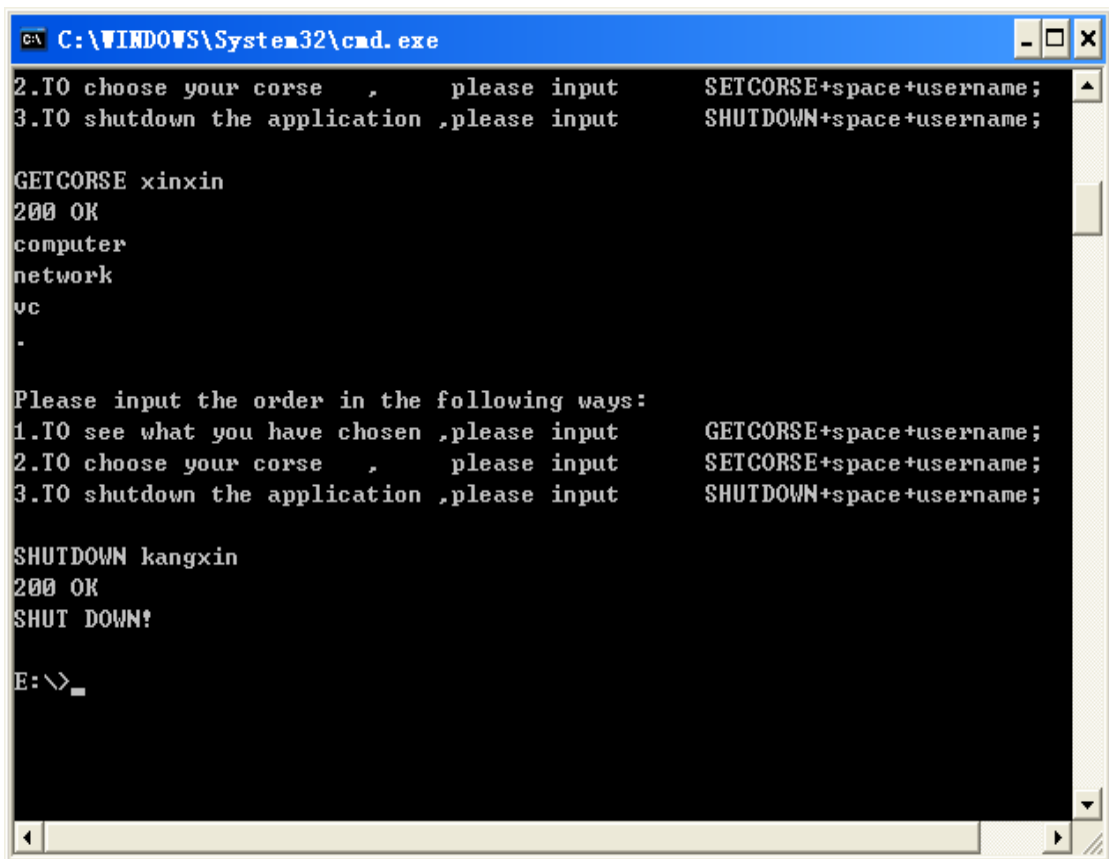
Please input the order in the following ways:
1.TO see what you have chosen ,please input      GETCORSE+space+username;
2.TO choose your corse ,      please input      SETCORSE+space+username;
3.TO shutdown the application ,please input      SHUTDOWN+space+username;
```

服务器端显示如下:



```
C:\WINDOWS\System32\cmd.exe - server.exe 1166
xinxin
get is ready
xinxin
GETTING COURSE
GET OK
GET OK
GET OK
GET OK
GETCORSE xinxin
xinxin
get is ready
xinxin
GETTING COURSE
GET OK
GET OK
GET OK
GET OK
```

8、输入 SHUTDOWN 命令和特权用户名 (该程序的设置的特权用户为 kangxin), 可以关闭远程服务器



```
C:\WINDOWS\System32\cmd.exe
2.TO choose your corse , please input SETCORSE+space+username;
3.TO shutdown the application ,please input SHUTDOWN+space+username;

GETCORSE xinxin
200 OK
computer
network
vc
-

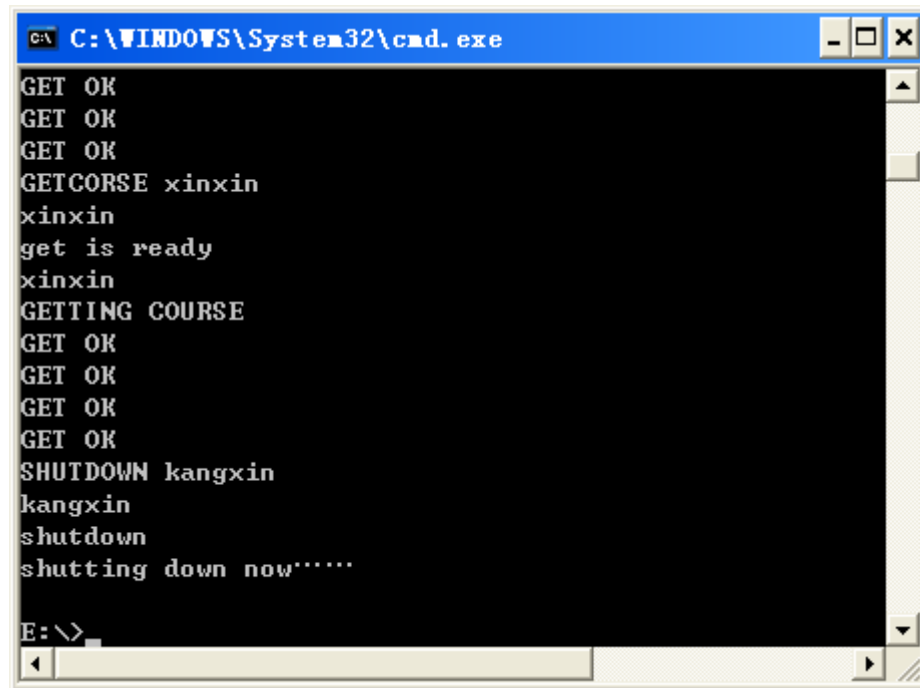
Please input the order in the following ways:
1.TO see what you have chosen ,please input GETCORSE+space+username;
2.TO choose your corse , please input SETCORSE+space+username;
3.TO shutdown the application ,please input SHUTDOWN+space+username;

SHUTDOWN kangxin
200 OK
SHUT DOWN!

E:\>
```

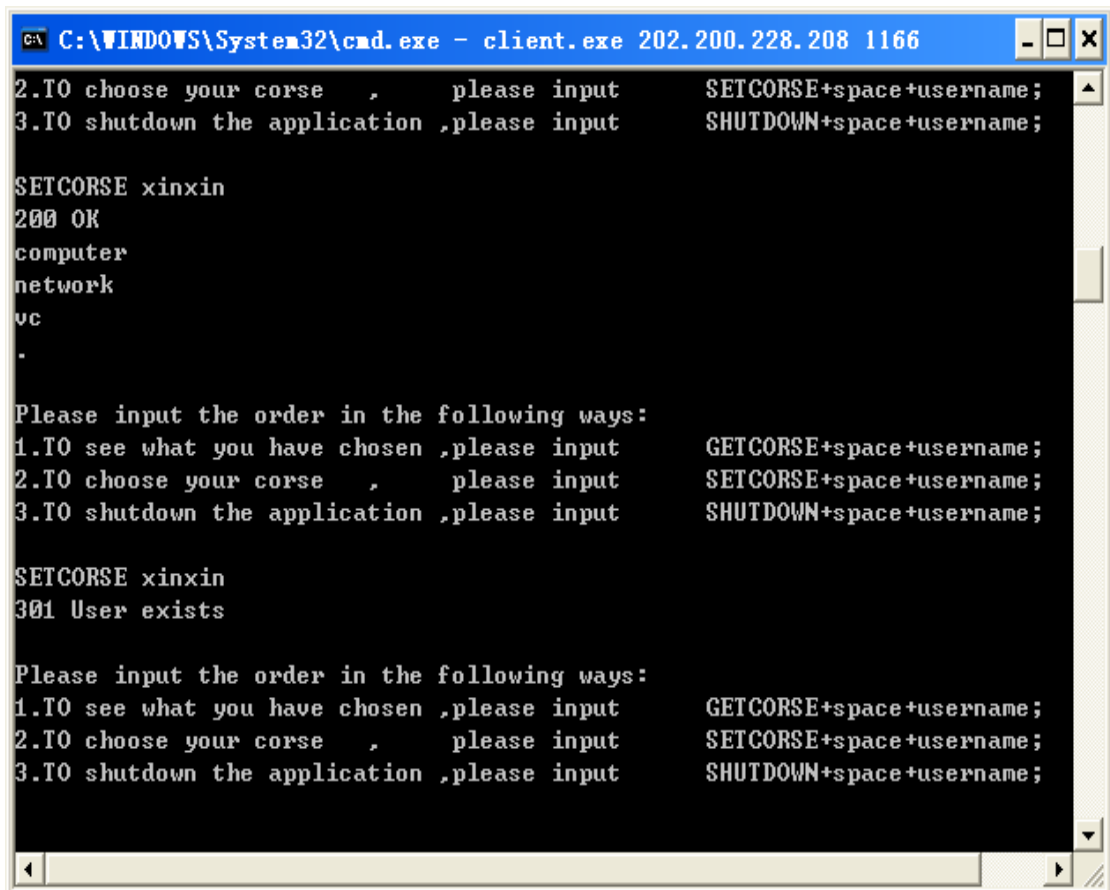


服务器端显示如下:



```
C:\WINDOWS\System32\cmd.exe
GET OK
GET OK
GET OK
GETCORSE xinxin
xinxin
get is ready
xinxin
GETTING COURSE
GET OK
GET OK
GET OK
GET OK
SHUTDOWN kangxin
kangxin
shutdown
shutting down now.....
E:\>
```

9、输入 SETCORSE 命令和用户名，用户名若存在，则显示 301 user exists，表示不可以选课



```
C:\WINDOWS\System32\cmd.exe - client.exe 202.200.228.208 1166
2.TO choose your corse , please input SETCORSE+space+username;
3.TO shutdown the application ,please input SHUTDOWN+space+username;

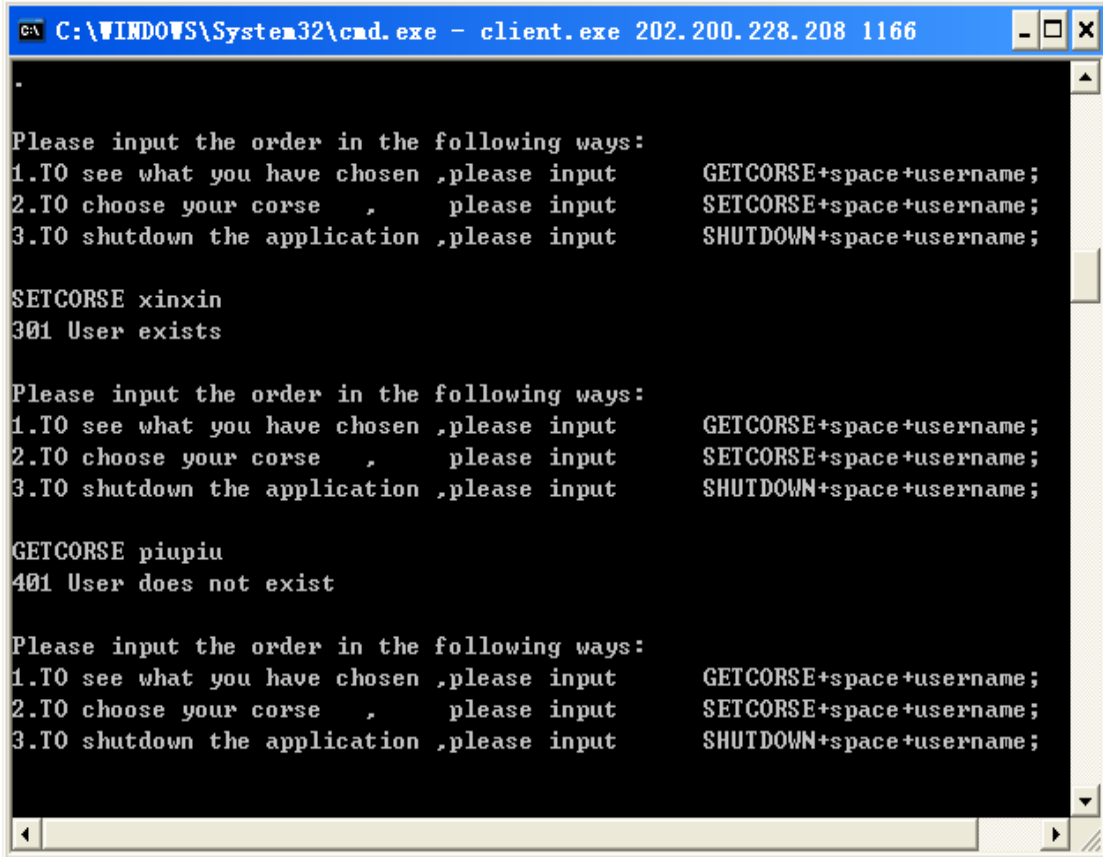
SETCORSE xinxin
200 OK
computer
network
vc
.

Please input the order in the following ways:
1.TO see what you have chosen ,please input GETCORSE+space+username;
2.TO choose your corse , please input SETCORSE+space+username;
3.TO shutdown the application ,please input SHUTDOWN+space+username;

SETCORSE xinxin
301 User exists

Please input the order in the following ways:
1.TO see what you have chosen ,please input GETCORSE+space+username;
2.TO choose your corse , please input SETCORSE+space+username;
3.TO shutdown the application ,please input SHUTDOWN+space+username;
```

10、输入 GETCORSE 命令和用户名，用户名若不存在，则显示 401 user does not exists，表示用户未选课



```
C:\WINDOWS\System32\cmd.exe - client.exe 202.200.228.208 1166

Please input the order in the following ways:
1.TO see what you have chosen ,please input      GETCORSE+space+username;
2.TO choose your corse ,      please input      SETCORSE+space+username;
3.TO shutdown the application ,please input      SHUTDOWN+space+username;

SETCORSE xinxin
301 User exists

Please input the order in the following ways:
1.TO see what you have chosen ,please input      GETCORSE+space+username;
2.TO choose your corse ,      please input      SETCORSE+space+username;
3.TO shutdown the application ,please input      SHUTDOWN+space+username;

GETCORSE piupiu
401 User does not exist

Please input the order in the following ways:
1.TO see what you have chosen ,please input      GETCORSE+space+username;
2.TO choose your corse ,      please input      SETCORSE+space+username;
3.TO shutdown the application ,please input      SHUTDOWN+space+username;
```

## 八、感想及总结

首先，完成这个程序时，我感到十分高兴和自豪，首先通过自己的劳动和付出，我看到了成果和收获，这是我最大的收获。其次通过这个程序的编写，我学到了很多的东西，我对书上的知识有了更深入的了解。而且，我也学会了查阅资料和利用工具书，并且增强了我的实际编程能力。

本次网络编程花了不少时间，一开始主要是查阅资料和对这个程序框架理解，接下来是开始构思算法，并进行结构化的设计程序。本次遇到的最大问题是调试的问题，总是有一些小疏忽，导致程序不可以正常运行。然后细心调试，再使问题一一解决。比如，开始我用的  $j = -1$ ，然后用的 while 循环，在循环里  $j++$ ，结果程序就不对，然后我

用 do-while 循环，在循环里 j++，我的程序就可以正常运行了，因为，我数组不可以从 -1 开始执行，而我得判断条件是 while(strcmp(student[i].course[j], ".") != 0); 用 while 循环是先判断，所以就不对，而 do-while 是先执行，后判断，所以 j 就变成 0 了，就不存在问题了

通过这次程序设计，使我更加明白了做任何事都要有恒心和毅力，要坚持到底。同时也增加了我对编程的兴趣，增加了我对网络的兴趣，今后我会更加深入的学习网络知识，更加努力地学习程序语言，写出更好的程序。