|  |  |
| --- | --- |
| 标题 | 实践作业1：简单的网络应用程序编写 |
| 发布时间 | 2016年10月26日 |
| 截止时间 | 2016年11月13日 |
| 评分方式: | 打分制:100.0分 |
| 作业内容 | 利用socket编写一个简单的网络应用程序，获取服务器当前的时间和日期。  说明与要求：  1）对客户与服务器之间使用的协议进行设计。  2）分别采用流式套接字和数据报套接字进行实现。  3）可以使用C/C++或Java编程语言。  4）提交报告，对以上内容进行总结和分析（包括程序的结构和说明）。 |

**信息安全 1410658 杨旭东**

**实验目的：**

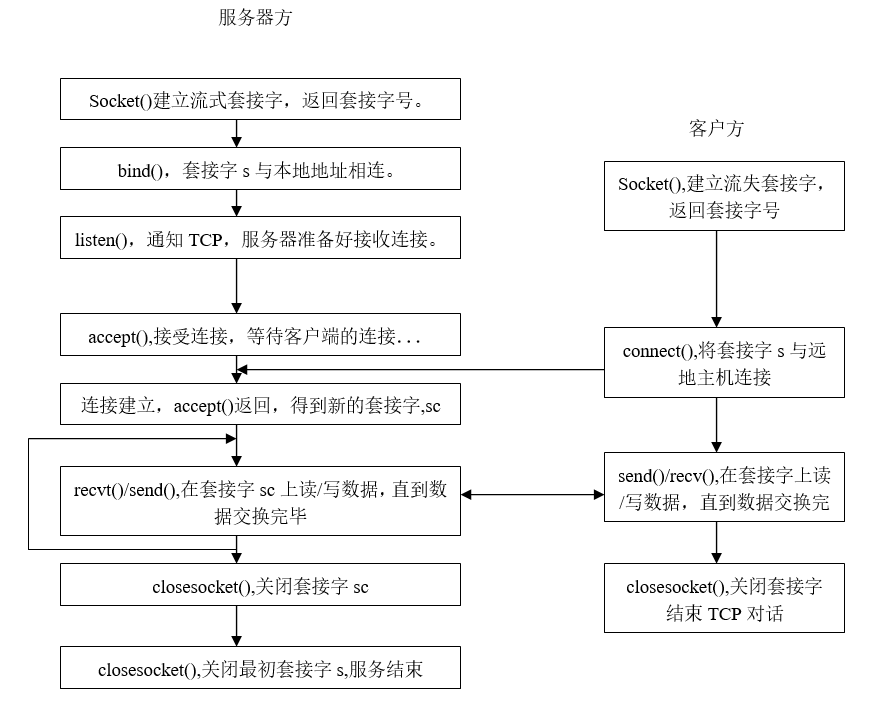
1、了解TCP与UDP协议，以及它们之间的区别。

2、了解客户/服务器模型原理。

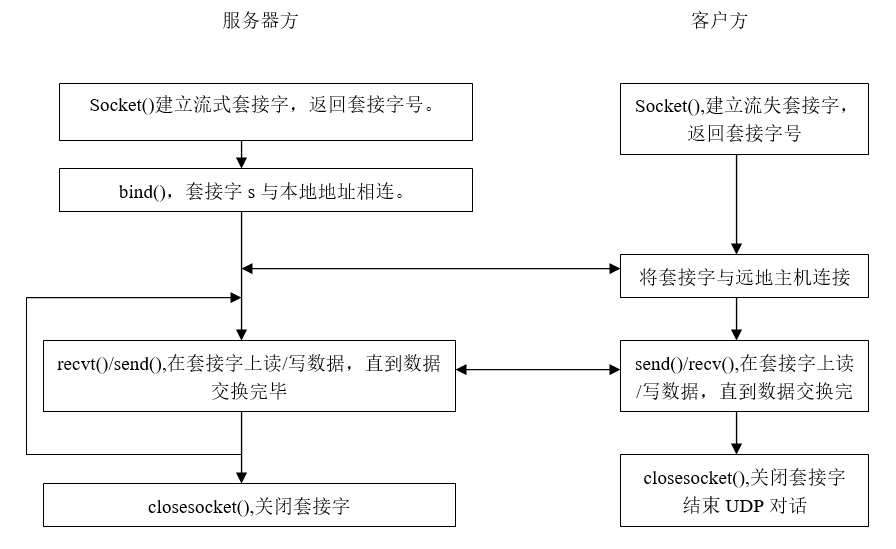
3、熟悉Socket编程原理，掌握简单的套接字编程。

**实验原理：**

1、编写用TCP协议实现的Client端和Server端程序并调试通过。

程序流程如图：

2、编写用UDP协议实现的Client端和Server端程序并调试通过。



**实验过程：**

本作业使用C++MFC完成，使用CAsyncSocket实现TCP和UDP模型。



**1.TCP模型**

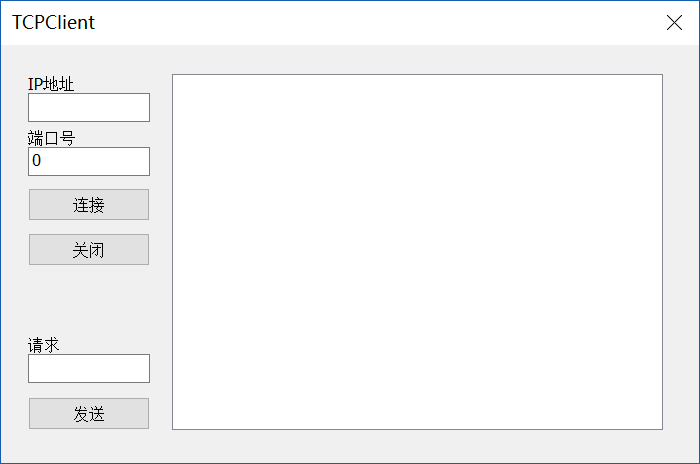
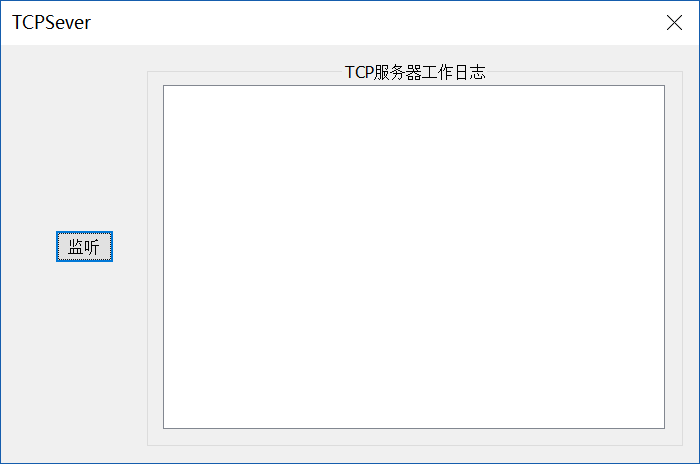
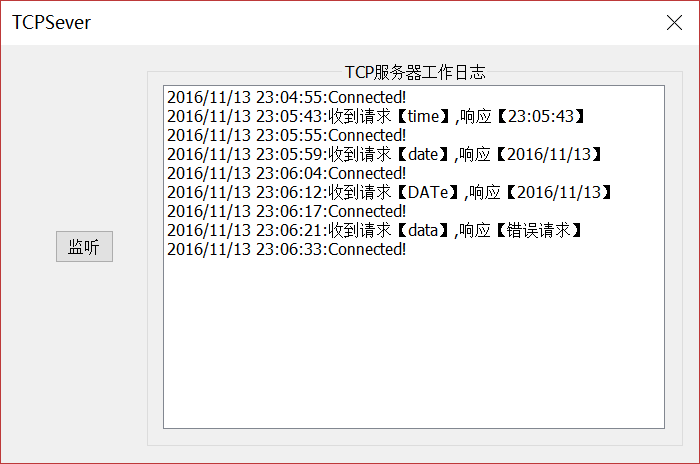
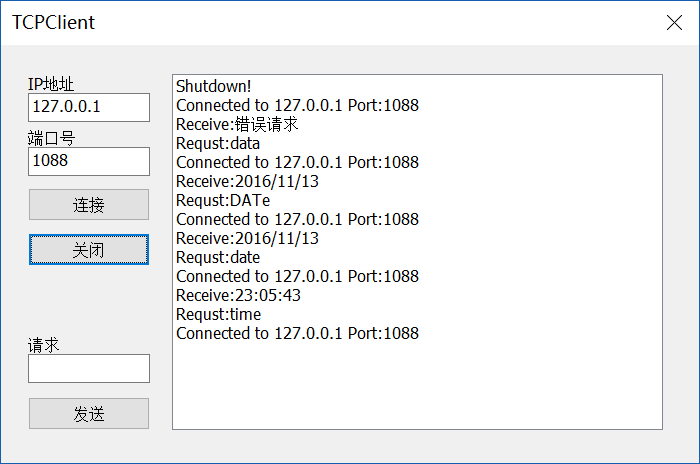
实现服务器需要两个派生自CAsyncSocket类的子类，一个用于Accept，一个用于Process。具体实现如下:

CListenSocket继承CAsyncSocket,重载OnAccpet,在OnAccept中Accept(\*processSocket),然后在CprocessSocket中处理。

CProcessSocket继承CAsyncSocket,重载OnSend OnReceive OnClose，通过AsyncSelect来切换读写的操作，在OnClose中delete本身（因为在accept中是new来的，不然会内存泄露)。

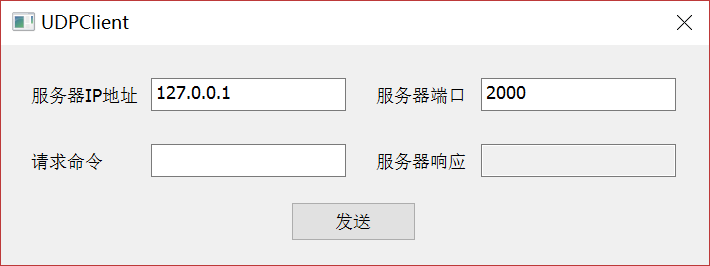
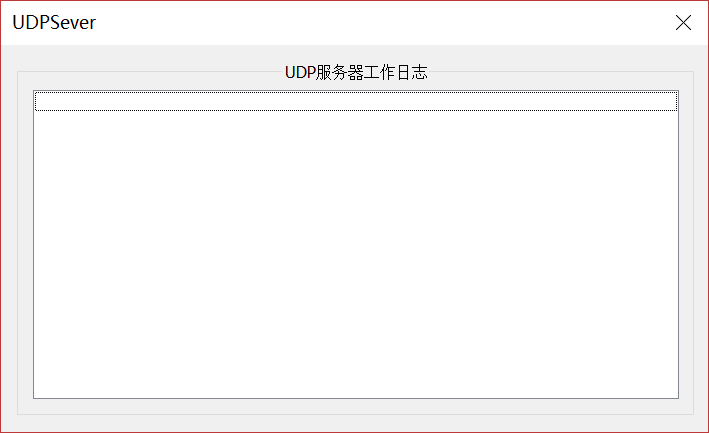
实现客户端只需要一个继承自CAsyncSocket的类，假设叫CConnectSocket,需要重载OnConnect OnSend OnReceive OnClose. OnConnect主要是为了收到接通时的事件，处理errorcode，其他跟服务器类似。

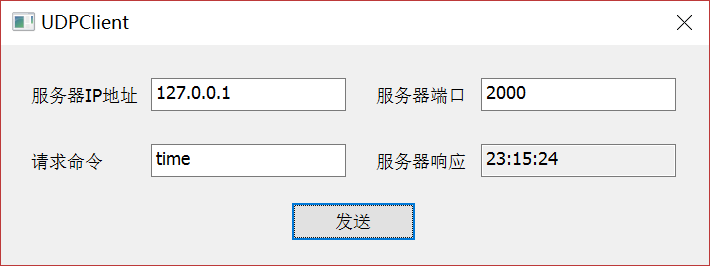
使用时，服务器用Create 并listen，客户端Create 然后Connect进入使用循环。

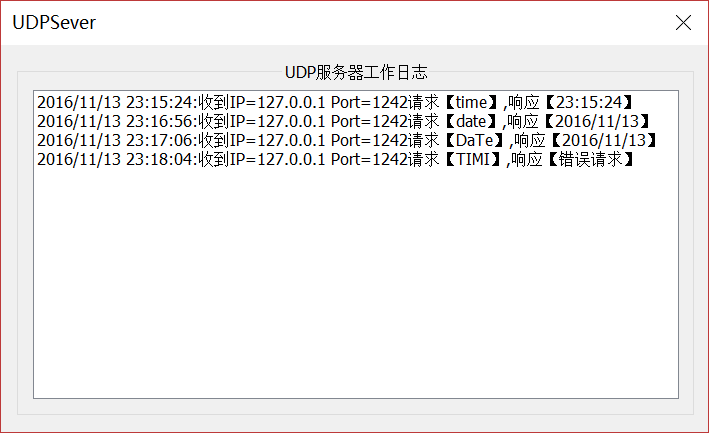
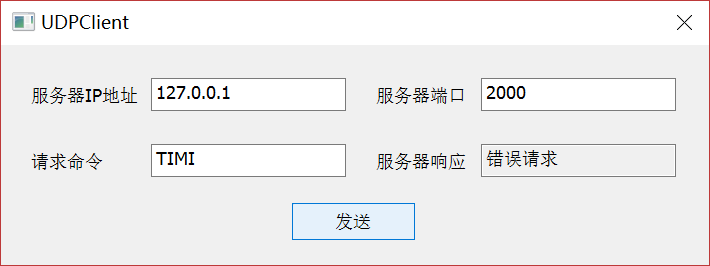
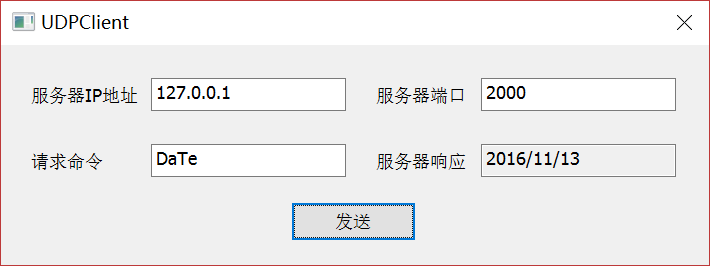
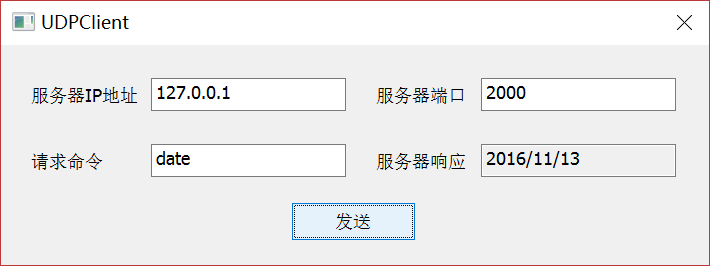
首先打开客户端和服务器界面，点击服务器监听，客户端连接（直接点击连接即可，默认IP为本机，端口号1088）后建立连接，发送请求，服务器返回请求，客户端接收，关闭连接，下次发送请求需要重新连接。点击关闭可以中断连接。客户端服务器均有日志记载。演示如图：

**2.UDP模型**

UDP模型比TCP要简单，不分服务器端跟客户端。也可以称为是对等的，都使用相同的派生于CAsyncSocket的类，这里假设为CupdSocket,重载OnReceive，

使用时，直接创建Create,然后用SendTo发送数据，接受数据都在OnReceive中，不需要AsyncSelect来切换。

同样打开客户和端服务器界面，自动建立连接（默认IP地址为本机，服务器端口号2000，客户端端口号！！！！）客户端发送请求，服务器响应，客户端接收。服务器记录日志。演示如图：



至此，实验完成。