## 网络编程课后习题作业

## 第一章

1. 简述分散式、集中式和分布式通信模型的特点：

分散式:各自独立的系统之间没有资源或信息的交换或共享。存在大量数据冗余且维护成本高,已被淘汰。

集中式:数据与管理功能都集中在单台计算机上,用户仅能通过终端来操作。优点就是硬件成本低、无冗余与不一致性问题,缺点就是可靠性与灵活性低,反应慢。分布式:就是分散式系统与集中式系统的混合体,就是目前比较完善的解决方案。

1. 简述C/S和B/S的优缺点：

C/S就是一种胖客户端应用程序编程架构,其主要工作都在客户端运行。优点就是可以充分利用本地计算机的性能优势,缺点就是需要在本机安装客户端软件。

B/S使用HTTP进行通信。优点就是单台计算机可以访问任何Web服务器,缺点就是B/S模式有沙盒限制。

1. 什么是套接字？套接字有哪几种类型？：

套接字就是支持TCP/IP网络通信的基本操作单元,就是不同主机间的进程进行通信的端点。在一个套接字实例中,既保存了本机的IP地址与端口,也保存了对方的IP地址与端口,同时也保存了双方通信采用的网络协议等信息。

套接字有3种类型:流式套接字、数据报套接字与原始套接字。

## 第三章

1. 简要回答下列问题
2. 进程和线程有什么区别？为什么要用多线程？多线程适用于哪种场景？
3. 前台进程和后台进程有什么区别和联系？如何判断一个线程属于前台进程还是后台进程？如何将一个线程设置为后台线程？
4. 进程就是正在执行的程序,线程就是某个进程中的一个或多个执行流。

多线程可让多个任务同时执行。

当执行需要较长时间才能完成的连续操作时,或者等待网络或其它I/O设备相应时,都可以使用多线程技。

1. 一个线程要么就是前台线程要么就是后台线程。

区别是:后台线程不会影响进程的终止,而前台线程则会影响进程的终止。

1. 什么是同步？为什么需要同步？C#提供了什么语句可以简单地实现代码同步？

执行某语句时,在该语句完成之前不会执行其后面的代码,这种执行方式称为同步执行。

当并行执行的多个线程同时访问某些资源时,必须考虑如何让多个线程保持同步。同步的目的就是为了防止多个线程同时访问某些资源时出现死锁与争用情况。

C#提供的lock语句可以简单地实现代码同步。

1. 什么是线程池？使用线程池有什么好处？

线程池就是在后台执行任务的线程集合。

好处有:如当某个线程无法进入线程池执行时先将其放入等待队列,自动决定用哪个处理器执行线程池中的某个线程,自动调节这些线程执行时的负载平衡问题等。另外,线程池总就是在后台异步处理请求的任务,而不会占用主线程,也不会延迟主线程中后续请求的处理。

## **第四章**

1. 什么是编码？什么是解码？为什么要对字符进行编码和解码？.NET框架提供了哪些用于字符编码和解码的类？

将字符序列转换为字节序列的过程称为编码;

当这些字节传送到网络的接收方时,接收方再将字节序列转换为字符序列,这种过程称为解码。

在网络通信中,很多情况下通信双方传达的都就是字符信息。但就是,字符信息并不能直接从网络的一端传递到另一端,这些字符信息首先需要被转换成一个字节序列,然后才能在网络中传输。因此,发送方需要进行编码,相应的接收方就要执行解码。

.NET框架中一般常用Encoding类实现编码与解码。Encoding类位于System、Text命名空间下,该类主要用于对字符集进行编码与解码以及将一种编码格式转换为另一种编码格式。如果有更详细的需求可以还使用与Encoding类位于同一命名空间下的Encoder与Decoder类来进行编码与解码。

1. .NET提供的从Stream类继承的数据流都有哪些？

文件流(FileStream)、内存流(MemoryStream)、网络流(NetworkStream)加密流(CryptoStream)

## 第五章

1. 仅包含async和await关键字的异步方法与用Task.Run调用的异步方法有何不同？

async和await关键字是C# 5.0提供的功能，仅包含async和await关键字的异步方法不会创建新线程，它只是表示在当前线程中异步执行指定的任务。而Task.Run方法是.NET框架4.5提供的功能，它会在线程池中用单独的线程执行某个任务

1. 把普通方法和异步方法作为任务来执时，调用方法有何不同？

普通方法要用Task.Run方法去调用，或者用Task、Task<TResult>类的构造函数显示创建Task实例，然后再启动。异步方法不需要用Task.Run方法去调用

1. Action和Func委托有什么不同？

Action委托封装了不带返回值的方法（有0～16个输入参数，返回类型为void），Func委托封装了带返回值的方法（有0～16个输入参数，返回类型为TResult）

1. 有几种创建任务的方式？

（1）利用Task.Run方法隐式创建和执行任务；

（2）利用async和await关键字隐式创建异步任务；

（3）利用WPF控件的调度器隐式创建和执行任务；

（4）通过显式调用Task或Task<TResult>的构造函数创建任务。

## 第六章

1. TPL支持哪些并行方式？

TPL支持数据并行，任务并行和并行查询（PLINQ）。

3、简述Parallel帮助器类有哪些？功能分别有什么？

帮助器类包括ParallelOptions、ParallelLoopState、ParallelLoopResult、CancellationToken和CancellationTokenSource。

ParallelOptions类用于为Parallel类的方法提供操作选项，常用属性有：CancellationToken、MaxDegreeOfParallelism、TaskScheduler。

ParallelLoopState类用于将Parallel循环的迭代与其他迭代交互。常用属性和方法有：IsExceptional属性、IsStopped属性、Break方法、Stop方法。

ParallelLoopResult类用于提供Parallel循环的完成状态，常用属性有：IsCompleted、LowestBreakIteration。

CancellationTokenSource用于创建取消通知，称为取消源。

CancellationToken结构用于传播应取消操作的通知，称为取消令牌。

1. 使用Parallel.Invoke方法时，为了能让Action与WPF界面交互，要注意什么问题？

不能使用默认的任务调度程序，而是通过设置并行选项，将任务调度程序与WPF当前同步上下文关联起来，相关代码如下:

ParallelOptions options = new ParallelOptions();

options.TaskScheduler = TaskScheduler.FromCurrentSynchronizationContext();

## 第七章

1. 简要介绍WCF服务的承载方式及其特点

WCF服务的承载方式包括：利用IIS或者WAS承载、利用Windows服务承载、自承载。其特点如下。

（1） 用IIS或者WAS承载：这是最常用的一种承载方式，特点是部署简单，方便。

（2） 用Windows服务承载：用【WCF服务库】模板创建的WCF服务利用Windows进程去承载。

（3） 自承载：指开发人员自己编写代码实现承载WCF的工作。优点是实现灵活，需要的环境支持最小。缺点是高级宿主管理功能需要自己写代码去实现。

1. 什么是服务协定？什么是数据协定？分别用哪些特性声明服务协定和数据协定？

服务协定是指WCF对客户端公开哪些服务。WCF服务端通过服务协定向客户端公开以下内容：操作方法、消息交换模式、采用的通信协议以及序列化格式。服务协定用ServiceContract特性和OperationContract特性来共同声明。

数据协定描述的是消息中包含哪些数据，即规定哪些数据能够被序列化为XML传输到客户端。数据协定用DataContract特性和DataMember特性来共同声明。

## 第八章

1. 编写基于C/S的HTTP应用程序时，有哪几种实现技术？各自的特点是什么？

(1)．用可插接式协议实现

可插接式协议用WebRequest类和WebResponse类来实现。这两个类是各种与具体的通信协议相关的类的基类，提供了上传、下载等基本方法。或者说，不论采用的是哪种通信协议，都可以用这两个对来实现。

(2)．用HttpWebRequest和HttpWebResponse实现

HttpWebRequest类和HttpWebResponse类是针对HTTP而提供的，分别从WebRequest类和WebResponse类继承而来。

(3)．用WCF实现

用WCF和基于任务的编程模型实现HTTP应用编程是面向服务编程中建议的做法。

前两种方式都是传统的编程模型，在实际项目中，如果用传统的编程模型来实现，除了业务处理之外，很多细节也都需要程序员自己去完成，例如负载平衡、网络监视、安全管理、防范攻击等。而用WCF来实现，程序员只需要处理业务逻辑即可，其他工作让WCF内部去完成就行了。

1. WCF客户端和服务端的消息交换模式有哪些？

请求/应答模式、单向模式、双工模式。

## 第九章

1. TCP有哪些主要特点？

（1）一对一通信。一旦通信双方建立了TCP连接，连接中的任何一方都能向对方发送数据和接收对方发送来的数据。每个TCP连接只能有两个端点，而且只能一对一通信。

（2）安全顺序传输。通过TCP连接传送的数据，能保证数据无差错、不丢失、不重复地准确到达接收方，并且保证各数据到达的顺序与数据发出的顺序相同。

（3）通过字节流发送数据。利用TCP传输数据时，数据以字节流的形式进行传输。

（4）传输的数据无消息边界。

1. 解决TCP的无消息边界问题有哪些常用的方法？

（1）发送固定长度的消息。

（2）将消息长度与消息一起发送。

（3）使用特殊标记分隔消息。

1. 简述基于任务的异步TCP编程和直接用Thread实现的主要区别

直接用Thread来实现TCP编程时，是以多线程为目标来实现的，需要开发人员自己管理多线程，因此开发人员必须熟练掌握多线程实现的所有细节，才能编写出实际的TCP应用程序。而基于任务的异步TCP编程是通过任务来实现TCP编程，用多任务实现时，即使开发人员对多线程、线程池以及资源冲突和负载平衡等所有技术实现细节不太熟悉，一样可以快速编写出实际的TCP应用程序，而且程序的健壮性比直接用多线程来实现要高得多。

1. 简要回答用WCF编写TCP服务器端和客户端程序的一般步骤

使用WCF编写TCP服务器端应用程序的一般步骤如下。

1）创建一个WPF应用程序，在项目中添加WCF服务。

2）在WCF服务接口文件中定义服务端与客户端通信的协定，并实现定义在服务端需要实现的协定方法。

3）修改服务端配置文件，并启动WCF服务，等待客户端调用并与客户端通信。

4）根据实际情况确定是否关闭WCF服务。

使用WCF编写TCP客户端端应用程序的一般步骤如下。

1）创建一个WPF应用程序，添加服务引用。

2）实现WCF服务定义在客户端需要实现的协定方法。

3）调用WCF服务，与服务端进行通信。

## 第十章

1. UDP和TCP的主要区别有哪些？

UDP是简单的、面向数据报的无连接协议，提供了快速但不一定可靠的传输服务。与TCP一样，UDP也是构建于底层IP协议之上的传输层协议。与TCP相比，UDP具有以下特点。

（1）UDP不但支持一对一通信，而且支持一对多通信。

（2）UDP传输速度比TCP快。

（3）UDP有消息边界，使用UDP不需要考虑消息边界问题。

（4）UDP不确保数据的发送顺序和接收顺序一致。

（5）UDP可靠性不如TCP。

1. 什么是广播？什么是多路广播？两者有什么区别？

广播是指同时向多个设备发送消息，并且所有子网中的设备都可以接收到发送方发来的消息。每个广播消息包含一个特殊的IP地址。广播消息地址分为两种类型：本地广播和全球广播。本地广播可以向子网中的所有设备发送广播消息，其他网络不会受到本地广播的影响。

多路广播是将消息从一台设备发送到本网或全网内选择的设备子集上，即发送到那些加入指定组播组的设备上。

二者区别：广播只能向其所在的子网内发送消息，而且是向子网中的所有设备发送消息，没有目的性，不但造成了网络负载，而且资源消耗较高。而多路广播是将消息发送到加入到特定组播组内的机器上，消息有目的性，资源消耗不高。

1. 简要回答利用UdpClient对象加入和退出多播组的步骤

在UDP协议中，广播和组播的实现都是借助于特殊的IP地址实现的。特别是组播比较特殊。对于IPv4来说，多播是指在224.0.0.0到239.255.255.255的D类IP地址范围内进行广播。

加入多播组时，首选创建UdpClient对象，然后使用JoinMutiCastGroup方法加入组播组。当退出组播组时，使用UdpClient对象的DropMulticastGroup方法可以退出多播组，参数中指出要退出多播组的IPAddress对象。

1. 简要回答用WCF和UDP编写多播程序与用UdpClient类编写多播程序有哪些

主要的不同点？这两种实现方式中，各自的优缺点有哪些？

使用UDP编写多播程序时，均需要将消息发送到特殊的IP地址。使用UdpClient类编写多播程序时，需要先加入到特定的多播组，然后才能接收来自多播发送端发送的数据。具体实现时主要是利用UdpClient类的JoinMutiCastGroup方法加入组播组，发送端利用UdpClient对象的Send方法发送数据到组播地址，接收端利用UdpClient对象的Receive方法接收数据。若要退出组播组调用DropMulticastGroup方法即可。

使用WCF和UDP编写多播程序时，需要使用WCF服务定义和实现通信的协定，然后向组播地址发送消息，这样凡是实现WCF服务定义和实现通信协定的客户端均可以接收到消息。

使用UdpClient类实现时，主要借助多线程和多任务来实现，易于理解，但是需要考虑较多的实现细节。而WCF模式对UDP多播编程进行了进一步的封装，降低了编程的复杂度，更适合分布式环境下的UDP通信程序。