**C#网络编程第一套**

1. 选择题

1.下列描述中哪个不是UDP的特点(A)

A.传输消息无消息边界。.

B.可以以广播或组播的方式发送数据。

C.一种无连接的协议。

D.不保证数据到达的顺序与数据发出的顺序相同。

|  |
| --- |
| UDP特点：   （1）UDP可以一对多传输   （2）UDP传输速度比TCP快   （3）UDP有消息边界   （4）UDP不保证有序传输 UDP不确保数据的发送顺序和接收顺序一致。对于突发性的数据报，有可能会乱序。   （5）UDP可靠性不如TCPUDP不提供数据传送的保证机制。 |

2.为了确定是否是创建控件的线程访问该控件的对象，需要检查的属性是(B).

A. Visible B. InvokeRequired C. CanSelect D. Enabled

|  |
| --- |
| 默认情况下，.NET框架都不允许在一个线程中直接访问另一个线程中的控件。   为了解决死锁以及异步执行过程中的同步问题，WPF中的每个元素（包括根元素）都有一个Dispatcher属性。   要在后台线程中与用户界面交互，可以通过向WPF控件的Dispatcher注册工作项来完成。   注册工作项的常用方法有两种：Invoke方法和InvokeAsync方法。 |

3.什么是CPU调度和分配的基本单位（A)

A.线程 B.进程 C.程序 D.线程池

|  |
| --- |
| 线程   进从程序实现的角度来说，将一个进程划分为若干个独立的执行流，每个独立的执行流都称为一个线程。   一个进程中既可以只包含一个线程，也可以同时包含多个线程。  （1）线程是CPU调度和分配的基本单位。  （2）每个进程都有一个主线程。  （3）除了主线程以外，还可以给一个进程分配若干个子线程，从而达到多个任务并行执行的目的。 |

4.利用TCP进行通信时，发送方第一次发送的字符串为“4567”，第二次发送的字符串为“abcd”，接收方不可能出现的情况为(D)

A.第一次接收4567， 第二次接收abcd

B.第一次接收4567a，第二次接收bcd

C.第一次接收4567， 第二次接收abc,第三次接收d

D.第一次接收456， 第二次接收abcd

|  |
| --- |
| TCP是Transmission Control Protocol（传输控制协议）的简称，是TCP/IP体系中面向连接的运输层协议，在网络中提供双工和可靠的服务。  特点：  一对一通信。  安全顺序传输。  通过字节流收发数据。  传输的数据无消息边界  TCP是将数据组装为多个数据报以字节流的形式进行传输，因此可能会出现发送方单次发送的消息与接收方单次接收的消息不一致的现象。 |

5. Process类提供了获取本机和远程计算机上相关进程的方法，下列哪个方法用于获取本机所有的进程。(D).

A. GetProcessesByName B. GetProcessByID

C. GetProcessByName D. GetProcesses

|  |
| --- |
|  GetProcessById方法（静态方法）：通过进程Id创建新的Process组件，并将其与本地计算机上的进程资源关联。 GetProcessById最多只有一个Process实例。   GetProcesses方法（静态方法）：获取本机所有进程   GetProcessesByName方法（静态方法）：获取本机上特定名称的进程 |

6.下列说法不正确的是（B）

A.组播组是开放的，可以随时加入到组播组中，也可以随时离开。

B.组播消息不能穿越子网。

C.组播通信时不需要向组播组发起连接请求。

D.组播地址范围在224. 0.0.0- 239.255. 255. 255。

|  |
| --- |
| 由于广播是向子网中的所有计算机用户发送消息，没有目的性，不但增加了网络传输负担，而且资源消耗较高。组播的出现，较好地解决了这个问题。组播也叫多路广播。所谓组播是将消息从一台计算机发送到本网或全网内选择的计算机子集上，即发送到那些加入指定组播组的计算机上。组播组是开放的，每台计算机都可以通过程序随时加入到组播组中，也可以随时离开。   组播组是分享一个组播地址的一组设备。与IP广播类似，IP组播使用特殊的IP地址范围来表示不同的组播组。组播地址(1110开头的D类地址)是范围在224.0.0.0到239.255.255.255的D类IP地址。任何发送到组播地址的消息都会被发送到组内的所有成员设备上。组可以是永久的，也可以是临时的。大多数组播组是临时的，仅在有成员的时候才存在。用户创建一个新的组播组时只需从地址范围内选出一个地址，然后为这个地址构造一个对象，就可以开始发送消息了。  组播协议的优势：  组播协议的优势在于当需要将大量相同的数据传输到不通主机时，  1．能节省发送数据的主机的系统资源和带宽；  2．组播是有选择地复制给又要求的主机；  3. 组播可以穿越公网广泛传播，而广播则只能在局域网或专门的广播网内部传播；  4. 组播能节省网络主干的带宽； |

7.当利用FTP协议传输文本时，应采用哪种传输方式（）

A. ASCII . B.二进制 C.Unicode D.GB- 2312

|  |
| --- |
| FTP的文件传输方式有两种：  　1、ASCII模式。ASCII模式一般用来传输文本文件；  　2、binary（二进制）模式。除了文本文件外的其他文件用binary（二进制）模式。 |

8.发送电子邮件经常使用的协议是(B)

A.POP3 B.SMTP C.FTP D.P2P

|  |
| --- |
| POP3协议用于接收或下载邮件，默认端口110 SMTP协议用于传输或发送邮件，默认端口25   ftp文件传输协议使得[主机](http://baike.haosou.com/doc/5331327-5566564.html)间可以共享文件，默认使用TCP的21端口标识  P2P Peer-to-peer 是一类允许一组用户互相连接并直接从用户硬盘上获取文件的网络。 |

9.下列中那个不是进行邮件发送和接收时常用的类()

A.Attachment B.SmtpClient C.MailMessage D.FtpWebRequest

|  |
| --- |
| ftp文件传输协议使得[主机](http://baike.haosou.com/doc/5331327-5566564.html)间可以共享文件,所以FtpWebRequest不能进行邮件的发送和接受 |

10.下列描述不正确的是 (A)

A. P2P 是一种新的网络传输协议。

B. P2P 架构中若存在中央服务器，则该服务器起到促成各节点协调的功能。

C.常见的P2P 软件有PPLIVE、QQ、POPO、 BT。

D.服务器和客户端通信需要知道对方使用的IP地址和端口号。

|  |
| --- |
| P2P传输协议起源较早，常见新的协议有IPFS协议等 |

11.在FTP协议中建立控制连接时服务器使用的端口是(A)

A. 21 B.110 C.80 D.81

|  |
| --- |
| ftp文件传输协议使得[主机](http://baike.haosou.com/doc/5331327-5566564.html)间可以共享文件，默认使用TCP的21端口标识。  ssh 22  Secure Shell 安全外壳协议，专为远程登录会话和其他网络服务提供安全性的协议  http 80（8080） Hyper Text Transfer Protocol 超文本传输协议，用于网页浏览  DNS 53 Domain Name System 域名系统，域名解析  https 443 Hypertext Transfer Protocol Secure 超文本传输安全协议，用于安全浏览网页  www代理服务 8080 Apache Tomcat web server，进行网页浏览  smtp 25 Simple Mail Transfer Protocol 简单邮件传输协议  telnet 23 不安全的文本传送  pop3 110 Post Office Protocol |

12.下列说法不正确的是（A）

A.不对称加密算法能够使用流进行处理，所以速度较快。

B. HASH算法对不同的数据加密不可能得到相同的结果。

C.不对称加密算法用公钥加密的数据只能用私钥解密。

D.对称加密算法加密和解密数据使用同一个密钥。

|  |
| --- |
| 不对称加密（也叫公钥加密）特点  优点：密钥位数多，而且无法根据加密密钥推导出解密密钥。从而使保密强度更高。  缺点：加、解密速度慢，不适合加密大量数据。 |

13. 确实需要保存密钥时，比较安全的办法是应该保存在哪里（D)。

A.XML B.文本 C.程序中 D.密钥容器

|  |
| --- |
| 密钥容器   不论是对称加密还是非对称加密，都有如何保存密钥的问题。   比如我们常见的让用户自己记忆密码的办法，如果攻击者也知道了该密码，那么攻击者也一样可以用它来进入系统或者利用它来解密数据，这样一来，加密也就失去了意义。   所以，网络传输中一般不使用让用户记忆密码的办法，而是自动产生密钥。   同时还必须有一种办法，来确保密钥存储的安全性，这就是密钥容器的用途。 |

14.下面算法中不对称加密算法为()。

A.RC2 B.DES C.TripleDES D. RSA.

|  |
| --- |
| 常用的不对称加密算法   .NET框架提供了多种实现不对称加密算法的类，支持的不对称加密算法有：   （1）RSA算法   （2）ECC算法 |

15. XNA中通过内容管道加载的资源利用什么属性访问（A）

A.AssetName B.fileName C.Name D. File

|  |
| --- |
| 未学，略 |

1. 填空题

1. C类网络地址192. 168. 3. 5,使用子网掩码255. 255. 255. 0,则本地广播地址为()。

|  |
| --- |
| 计算机网络知识点，未学，略 |

2.如果一个线程t1在执行的过程中需要等待另一个线程t2结束后才能继续执行，可以在线程t1调用线程(t2)的(join)方法。

|  |
| --- |
| Thread类提供的常用方法   Start方法：启动线程   Join方法：将指定的线程合并到当前线程中，并阻止当前线程执行，直到指定的线程终止或经过了指定的时间为止   Sleep方法：将当前线程阻止指定的毫秒数，零(0)表示应挂起此线程以使其他等待线程能够执行   Abort方法：在调用此方法的线程上引发 |

3. (套接字)是网络通信的基本操作单元，(网络流)对象是专门用于网络通信的数据流。

|  |
| --- |
| 【问答题】什么是套接字？套接字有哪几种类型？  套接字是支持TCP/IP网络通信的基本操作单元，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地址和端口，也保存了对方的IP地址和端口，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、数据报套接字和原始套接字。  【网络流】  网络流（NetworkStream）  System.Net.Sockets命名空间下的NetworkStream类也是从Stream类继承而来的，利用它可以通过网络发送或接收数据。 |

4. FTP协议在建立数据连接时，支持两种工作模式(主动).（被动)。

|  |
| --- |
| 进行FTP协议的文件传输过程中，必须先掌握FTP协议的两种连接方式，才能更加顺畅的在代码中实现FTP服务器的开发或者FTP客户端的开发，其中两种工作方式的原理如下所示：  1、PORT方式（主动方式）：  客户端向服务器的FTP端口（默认是21号）发送连接请求，服务器接收连接，建立一条命令链路。当需要传送数据时，客户端使用并监听某临时端口，并在命令链路上用PORT命令告诉服务器端采用主动传输方式。于是服务器从20端口向客户端的临时端口发送连接请求，建立一条数据链路来传送数据。  2、PASV方式(被动方式)：  客户端向服务器的FTP端口（默认是21号）发送连接请求，服务器接收连接，建立一条命令链路。当需要传送数据时，客户端告诉服务器采用被动传输方式。于是服务器使用并监听某临时端口，并在命令链路上用PASV命令告诉客户端对应的端口号。最后，客户端向服务器的端口发送连接请求，建立一条数据链路来传送数据。 |

5.在SMTP 中，电子邮件由三部分组成: （信封）（首部）（正文）。

|  |
| --- |
| 对于SMTP协议C#进行了很多封装，使我们使用起来相当方便  SMTP是英文Simple Mail Transfer Protocol的缩写，意为简单邮件传输协议，默认端口为25  在SMTP协议中，电子邮件由三部分组成，信封、首部和正文。 |

6. .在XNA Framework 2. 0中有三种特殊类型的矩阵: ()()。

|  |
| --- |
| 未讲，略 |

7. P2P的设计模式有()()两类。

|  |
| --- |
| 未讲，略 |

1. 简答题

1.什么是数据完整性验证?简要回答如何实现数据完整性验证?

答：数据完整性:验证数据在传输过程中是否被修改(3分)

实现原理:验证数据完整性的实现原理是:发送方先使用Hash算法对数据进行Hash运算得到数据的Hash值,然后将数据和Hash值一块儿发送给接收方;接收方接收到数据和Hash值后，对接收的数据进行和发送方相同的Hash运算,然后将计算得到的Hash值和接收的Hash值进行比较，如果二者一致，说明收到的数据肯定与发送方发送的原始数据相同，从而说明数据是完整的。(3 分)

|  |
| --- |
| 第四章知识  PPT上关于此知识点的描述：在应用程序中，可以利用数字签名实现数据身份验证和数据完整性验证。数据身份验证是为了验证数据是不是持有私钥的人发送的；数据完整性验证则用于验证数据在传输过程中是否被修改过。 |

2.解决TCP协议的无消息边界问题一般有几种方法?它们适合哪些场

合?

答：1)发送固定长度的消息。适用于信息长度固定的场合。

2)将消息长度与消息一起发送。 此方法适用于任何场合。

3)使用特殊标记分隔消息。适用于消息本身不包含特殊标记的场合。6分(只答方法不答适用场合给4分)

|  |
| --- |
| 第九章的知识  PPT上关于此知识点的描述：  解决TCP无消息边界问题的办法  （1）发送固定长度的消息  这种办法适用于消息长度固定的场合。  （2）将消息长度与消息一起发送  一般在每次发送的消息前面用4个字节表明本次消息的长度，然后将其和消息一起发送到对方；对方接收到消息后，首先从前4个字节获取实际的消息长度，再根据消息长度值依次接收发送方发送的数据。  这种办法适用于任何场合。  （3）使用特殊标记分隔消息  使用特殊分隔符对消息进行分隔。  这种办法主要用于消息本身不包含特殊标记的场合。 |

3.简述P2P和传统的C/S模式相比有哪些优缺点?

答：在传统的“服务器/客户端"应用系统中，客户端与服务器有明确的分界，而使用P2P技术实现的每个计算机节点既是客户端，也是服务器。(3分)采用这种方式，不会像C/S模式那样带来服务器网络带宽瓶颈，造成网络堵塞的问题。同时，P2P技术更加符合分布式计算的理念。(2分)但是P2P技术同样带来了网络安全问题(1 分)

|  |
| --- |
| 第一章的知识  PPT上关于此知识点的描述：  C/S（Client/Server）也叫C/S模式、C/S架构或C/S模型，它是在分布式的基础上进一步抽象出来的编程模型。  C/S模式及其特点：  胖客户端应用程序编程架构，其主要工作都在客户端运行；例如QQ、飞信、360安全卫士等客户端软件。  C/S既适用于实际的应用程序，又适用于真正的计算机部署。从程序实现的角度来说，客户端和服务端实际是计算机上的两个进程的交互。服务端进程逐一等待并处理客户端请求。运行服务端进程的计算机系统一般通过所提供的服务来命名。  P2P  　　对等连接(peer-to-peer，简写为 P2P)是指两个主机在通信时并不区分哪一个是服务请求方还是服务提供方。  　　只要两个主机都运行了对等连接软件（P2P 软件），它们就可以进行平等的、对等连接通信。  　　双方都可以下载对方已经存储在硬盘中的共享文档。  　　对等连接方式的特点  　　　　对等连接方式从本质上看仍然是使用客户服务器方式，只是对等连接中的每一个主机既是客户又同时是服务器。  　　　　例如主机 C 请求 D 的服务时，C 是客户，D 是服务器。但如果 C 又同时向 F提供服务，那么 C 又同时起着服务器的作用。 |

4.什么是线程同步?如何实现线程同步?

答:线程同步:指多个线程之间存在先后执行顺序的关联关系(3 分)lock语句可以有效地实现同步，即将代码段(语句块)标记为临界区。它能确保当一个线程位于代码的临界区(可以理解为一段代码)时，另一个线程不进入临界区。如果其他线程试图进入锁定的代码段，则它将一直等待(即被阻塞)，直到锁定的对象被释放以后才能进入临界区。(3分)

|  |
| --- |
| （1）用volatile修饰符锁定公共或私有字段。   private static volatile bool isStop = false;   public static bool IsStop   {   get { return isStop; }   set { isStop = value; }   }   利用volatile修饰符可直接访问内存中的字段，而不是将字段缓存在某个处理器的寄存器中。这样做的好处是所有处理器都可以访问该字段最新的值。   （2）用Interlocked类提供的静态方法锁定局部变量。   System.Threading.Interlocked类通过加锁和解锁提供了原子级别的静态操作方法，对并行执行过程中的某个局部变量进行操作时，可采用这种办法实现同步。   int num = 0;   Interlocked.Increment(ref num); // 将 num 的值加 1   Interlocked.Decrement(ref num); // 将 num 的值减 1   （3）用lock语句锁定代码块   C#提供了一个lock语句，该语句能确保当一个线程完成执行代码块之前，不会被其他线程中断。被锁定的代码块称为临界区。   lock语句的实现原理是进入临界区之前先锁定某个私有对象（声明为private的对象），然后再执行临界区中的代码，当代码块中的语句执行完毕后，再自动解除该锁。 |

5.简述POP3工作原理。

|  |
| --- |
| 未讲，略 |

1. 程序填空题

1.现有一个Windows应用程序，利用TCP实现服务端功能。下面列出了相应的事件及部分代码，请在下列选项中选择最确切的答案，将完成事件功能的代码填写完整。(10 分)

A Start() B AcceptTcpClient() C Begin() D IPHostEntry E Dns

F AcceptSoket () G sr. ReadLine() H sr. ReadString()

I client. GetStream() J GetValue() K Accept () L newNetWorkStream ()

(1)单击启动监听，则在本机第-一个可用IP地址、端口3035.上启动TCP监听，等待客户端的连接，每当接收到一个客户端连接请求后，启动一个线程负责接收并处理该客户端发送过来的信息。

private TcpListener myListener ;

private void buttonl\_ Click (object sender, EventArgs e)

String name = Dns. GetHostName ();

//获取本机所有IPAddress

IPAddress[] al1IP =\_\_\_\_\_①\_\_\_\_\_\_\_ GetHos tAddresses (name) ;

IPAddress ip = allIP[0];

myListener = new TcpListener (ip, port)

//启动监听

myListener. \_\_\_\_\_\_\_\_\_\_②\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Thread threadListenClient = new Thread (ListenClientConnet);

threadListenClient. Start() ;

}

(2) ListenClientConnet方法负责接收客户端的连接请求。

private void ListenClientConnect ()

{

while (true)

{

TcpClient newClient = null:

try

{

//等待用户进入

newClient = myListener. \_\_\_\_\_③\_\_\_\_\_\_\_\_

}

catch

{ break;}

Thread receiveDataThread = new Thread (ReceiveData) ;

receiveDataThread. Start (newClient) ;

}

}

(3)补充负责接收数据的ReceiveData方法。

private void ReceiveData (object obj)

{

TcpClient client = (TcpClient )obj;

bool exitWhile = false;

while (exitWhile == false)

{string receiveString = null;

try

{

//方法获取网络流

NetWorkStream stream =\_\_\_\_\_\_④\_\_\_\_\_\_

//利用StreamReader对网络流进行封装

StreamReader sr = new StreamReader (stream) ;

//利用sr读取数据

receiveString = \_\_\_\_\_\_⑤\_\_\_\_\_\_

}

catch

{service. SetListBox("接收数据失败”);}

}

答:① E ② A ③ B ④ I ⑤ G

|  |
| --- |
| 第一章编程题 |

2.现有一个Windows应用程序,该程序利用组播技术实现网络会议讨论。该会议使用组播组地址为224. 0.0.1,通讯端口为3035。下面是完成此功能的核心代码框架，在下列选项中选择最确切的答案，将完成事件功能的代码填写完整。(10 分)

A. UTF8 B GetBytes C ip D iep E ref remote

F remote G IPAddress H GetValue I ASCII J GetString

(1)将SendToAll方法填写完整。该方法实现加入到组播组中，并向组

播组发送组播消息的功能。

private void SendToAll (String strMessage)

{

UdpClient myUdpClient = new UdpClient () ;

IPAddress ip = IPAddress. Parse("224. 0.0.1");

myUdpClient. EnableBroadcast = true;

//实例化通讯用端点

IPEndPoint iep = new IPEndPoint (ip, 3035) ;

//加入组播组

udpClient. JoinMuticastGroup (\_\_\_\_①\_\_\_\_\_);

udpClient. Tt1 = 30;

//将strMessage转换为UTF8字节数组

Byte[] bytes= Encoding. UTF8. \_\_\_\_②\_\_\_\_(strMessage)

try

//向子网发送信息

udpClient. Send (bytes,\_ bytes. Length, \_\_\_\_\_③\_\_\_\_);

}

catch (Exception err)

{

MessageBox. Show(err. Message，” 发送失败"):

}

finally

{ myUdpClient. Close(); }

}

(2) ReceiveData 方法负责在端口3035接收组播消息。

private void ReceiveData ()

{

//在本机指定的端口接收

udpClient = new udpClient (3035) ;

IPEndPoint remote = null;

while (true){

try

{

//接收从远程主机发送过来的信息:

byte[] bytes = udpClient. Receive(④);

//将接收到的字节数组转化为字符串;

String str = Encoding.⑤.GetString (bytes) ; .

AppendStr ing (string. Format(来自{0}: {1}", remote,str));

}

catch { break; }

}

}

答:① C ② B ③ D ④ E ⑤ A

|  |
| --- |
| 第十章编程题 |

3.(1).NET基类库使用的私钥算法类采用的加密模式是密码本模式还是密码块链模式?如果有两个内容相同的原始数据块，使用. NET提供的对称加密类，使用相同的私钥进行加密，那么得到的加密后的两个数据块是否相同?并给出理由。(4 分)

答:

密码块链模式、不相同、即使私钥相同，初始化向量相同，加密后结果还和前一块内容相关

(2)方法EnCryptText实现利用TripleDES算法对字符串的加密，下面

列出了加密方法的部分代码，选择最确切的答案，将完成事件功能的代码填写完整。(6 分)

A cryptoStream B memoryStream C ToArray ()

D ToCharArray() E Abort() F Close()

// toEncrypt待加密字节数组

//Key算法使用密钥

//IV算法使用向量

private byte[] EncryptText (BYTE[] toEncrypt ，byte[] Key, byte[] IV)

{

//创建一个内存流

MemoryStream memoryStream = new MemoryStream();

//使用内存流、私钥和IV创建加密流，补充第1个参数。

CryptoStream cryptoStream = new CryptoStream (

①

New TripleDESCryptoServiceProvider().CreateEncryptor(Key,IV),CryptoStreamMode. Write);

Try

{

////将字节数组写入加密流

cryptoStream.Write (toEncrypt, 0，toEncrypt. Length)

cryptoStream. FlushFinalBlock0);

byte [] encryptedBytes = memoryStream.

return encryptedBytes;

}

catch (CryptographicException err){

throw new Exception(' 加密出错:” + err. Message) ;

}

Finally{

//关闭加密流

cryptoStream.③\_；

memoryStream. Close() ;

}

答: ① B ② C ③ F

|  |
| --- |
| 第四章数据加密编程题 |

**C#网络编程第二套**

1. 选择题

1.下列关于进程和线程的描述中哪个是错误的(C)

A.线程池适用于需要多个线程而实际运行时间比较短的场合。

B.在C#中可使用lock语句实现线程同步。

C.执行Thread实例的Start方法就意味着线程实例立即执行。

D.一个线程要么是前台线程要么是后台线程。

|  |
| --- |
| **线程的生命周期经历的状态：**  **新建**：**new**创建线程对象时  **就绪**：调用**start()**方法时  **运行**：调用**run()**方法时  **阻塞**： 多种原因可导致阻塞  **死亡**：多种原因 |

2.在C#中,使用那个类表示网络通信时使用的端点

A. IPAddress B. IPEndPoint C. IPHostEntry D. Dns

|  |
| --- |
| 复杂的网络应用程序中，需要掌握几个最基本的类：  1、提供网际协议IP地址的IPAddress类, 提供了对IP地址的转换和处理功能。  2、包含IP地址和端口号的IPEndPoint类；IPEndPoint是与IPAddress概念相关的一个类，它包含应用程序连接到主机上的服务所需的主机和端口信息。它由两部分组成，一个是主机IP地址，另一个是端口号。  3、为Internet或Intranet主机提供信息容器的IPHostEntry类。IPHostEntry类将一个域名系统（DNS）的主机名与一组别名和一组匹配的IP地址关联。该类一般和Dns类一起使用。  DNS（Domain Name System，域名系统）是因特网的一项核心服务，它可以将域名和IP地址相互转换。  //定义远程计算机端点信息  IPEndPoint sender = new IPEndPoint(IPAddress.Any, 0）；  可知，网络通信时使用的端点是IPEndPoint |

3.关闭不带有Windows界面进程的方法为(B)

A. CloseMainWindow B. Kill C. Abort D . Join

|  |
| --- |
| C、D选项是线程管理中提供的方法，题目问的是进程。所以应该从A、B中选，A是关闭具有用户界面的进程方法，所以选B，强制终止进程。  进程终止：  Kill方法：强制终止进程   CloseMainWindow方法：关闭具有用户界面的进程   Close方法：释放与此组件关联的所有资源   HasExited属性：指示关联进程是否已终止   WaitForExit方法：设置等待关联进程退出的时间，并在该段时间结束前或该进程退出前，阻止当前线程执行。  线程管理（Thread类）   Thread类提供的常用方法   Start方法：启动线程   Join方法：将指定的线程合并到当前线程中，并阻止当前线程执行，直到指定的线程终止或经过了指定的时间为止   Sleep方法：将当前线程阻止指定的毫秒数，零(0)表示应挂起此线程以使其他等待线程能够执行   Abort方法：在调用此方法的线程上引发ThreadAbortException，以开始终止此线程的过程。调用此方法通常会终止线程。 |

4.利用FTP协议上传及下载图形图像类的文件时，为了保证数据不发生变化,选择的传送方式应为（）

A.二进制 B. ASCII C.主动模式 D.被动模式

|  |
| --- |
| A、B是传输方式，C、D是工作模式（第一套卷子里的填空题有）  FTP的文件传输方式有两种：  　1、ASCII模式。ASCII模式一般用来传输文本文件；  　2、binary（二进制）模式。除了文本文件外的其他文件用binary（二进制）模式。 |

5.对于网络地址192.168.3.0，使用子网掩码255.255.255.0。本地广播地

A.本地广播地址是192.168.255.255

B.本地广播地址是255.255.255.255

C.本地广播地址是192.168.3.255

D.本地广播地址是192.168.3.0

|  |
| --- |
| 下学期知识，应该不考 |

6.下列FTP命令中不需要建立数据连接的命令为()

A. LIST B. RETA C. STOR D. DELE

|  |
| --- |
| 没学过，应该不考 |

7..NetFramework中UdpClient类提供的加入组播组的方法为(A)

A. JoinMuticastGroup B. DropMuticastGroup

C. InvokeRequired D. Connect

|  |
| --- |
| A 加入多播组  利用UdpClient对象的JoinMulticastGroup方法可加入到指定的多播组中，例如：  //创建UdpClient的实例并设置本地监听的端口号  UdpClient udpClient=new UdpClient(8001);  udpClient.JoinMulticastGroup(IPAddress.Parse("224.0.0.1"));  B 退出多播组  利用UdpClient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。例如：  udpClient.DropMulticastGroup(IPAddress.Parse("224.100.0.1"));  补充 是否允许接收多播  利用UdpClient对象的MulticastLoopback属性可控制是否允许接收多播信息。该属性默认为true，即允许接收多播。  C C#中禁止跨线程直接访问控件，InvokeRequired是为了解决这个问题而产生的，当一个控件的InvokeRequired属性值为真时，说明有一个创建它以外的线程想访问它。  D Connect方法  **建立默认远程主机。**因为UDP是无连接的，所以不会阻止调用该方法的线程。另外，如果打算接收多路广播数据报，不要调用Connect方法，否则从指定的默认地址以外的地址到达的任何数据报都将被丢弃。但是，如果在调用Send方法时指定了IPAddress.Broadcast，则可以将数据广播到默认的广播地址255.255.255.255 |

8.接收邮件经常使用的协议是(C)

A. TCP协议 B. UDP协议 C. POP3协议 D. SMTP协议

|  |
| --- |
| POP3协议用于接收或下载邮件，默认端口110 SMTP协议用于传输或发送邮件，默认端口25  ftp文件传输协议使得[主机](http://baike.haosou.com/doc/5331327-5566564.html)间可以共享文件，默认使用TCP的21端口标识  P2P Peer-to-peer 是一类允许一组用户互相连接并直接从用户硬盘上获取文件的网络。 |

9.下列描述中错误的是(D)

A.对等模式 B.设计模式可分为单纯型架构和混合型架构两种

C.常见的P2P应用有电驴、BT、QQ、迅雷

D. P2P是一种新的协议

|  |
| --- |
| P2P传输协议起源较早，常见新的协议有IPFS协议等 |

10..NET中提供的用来将数据流链接到加密转换流的类为(B)

A.MemoryStream B.CryptoStream C.FileStream D.StreamReader

|  |
| --- |
| 第四章提到常见的数据流  文件流（FileStream）、内存流（MemoryStream）、网络流（NetworkStream）、加密流（CryptoStream）  StreamReader和StreamWriter类（操作文本数据时，一般用StreamReader和StreamWriter类来实现。）  BinaryReader和BinaryWriter类（System.IO命名空间还提供了BinaryReader和BinaryWriter类以二进制模式读写流，更方便于对图像文件、压缩文件等二进制数据进行操作。） |

11.不对称算法也称什么算法(B)

A.私钥加密算法 B.公钥加密算法 C.公钥/私钥加密算法 D. Hash算法

|  |
| --- |
| 不对称加密   不对称加密也叫公钥加密。这种技术使用不同的加密密钥与解密密钥，是一种“由已知加密密钥推导出解密密钥在计算上是不可行的”密码体制。   不对称加密产生的主要原因有两个：   一是为了解决对称加密的密钥管理问题；   二是数字签名需要用不对称加密来实现。  A选项：  对称加密也称为私钥加密、采用私钥算法，加密和解密数据使用的是同一个密钥。 |

12.在利用TCP协议进行通信时，发送方第一次发送的字符串为“456”,第二次发送的字符串为“abc",则接收方无法出现的情况为(D)

A.第一次接收456,第二次接收abc

B.第一次接收456a，第二次接收bc

C.第一次接收45，第二次接收6abc

D.第一次接收456，第二次接收acb

|  |
| --- |
| TCP是Transmission Control Protocol（传输控制协议）的简称，是TCP/IP体系中面向连接的运输层协议，在网络中提供双工和可靠的服务。  特点：  一对一通信。  安全顺序传输。  通过字节流收发数据。  传输的数据无消息边界  TCP是将数据组装为多个数据报以字节流的形式进行传输，因此可能会出现发送方单次发送的消息与接收方单次接收的消息不一致的现象。 |

13.下列说法正确的是()

A.向本地广播地址发送的信息可以穿越子网

B.数字签名即实现数据身份认证

C.数字签名既可以实现数据身份验证和数据完整性验证

D. RSA算法为对称加密算法

|  |
| --- |
| A、第一套有这个选项，本地广播发送的地址不可以穿越子网。  B、除数据身份验证外还有数据完整性验证。  C、数字签名的实现原理，数字签名是利用私钥加密必须用公钥解密这个原理来实现的。在应用程序中，可以利用数字签名实现数据身份验证和数据完整性验证。数据身份验证是为了验证数据是不是持有私钥的人发送的；数据完整性验证则用于验证数据在传输过程中是否被修改过。  D、RSA算法：RSA算法不仅可用于不对称加密和解密，还能用它实现数字签名  PPT中的数字签名知识点：  通过Internet下载文件后，怎样知道下载的文件是否和原始文件完全相同呢？发送方通过Internet发送数据后，接收方如何验证接收的数据在网络传输过程中是否被修改过？   这就是数字签名的用途。   如果通信双方希望确保信息是来自对方而不是来自第三方，需要使用数字签名进行身份验证。   另外，数字签名还可以防止特定一方否认曾发送过的信息。  利用私钥加密必须用公钥解密这个原理实现。 |

1. 填空题

1.线程同步是指多个线程之间存在先后执行顺序的关联关系。

|  |
| --- |
| 同步执行和异步执行   一种是执行某语句时，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。   一种是执行某语句时，不管该语句是否完成，都会继续执行其后面的语句，这种执行方式称为异步执行。 |

2.利用Socket实现面向连接的通信时，应使用流套接字。

|  |
| --- |
| 流式套接字：实现面向连接的TCP通信  数据报套接字：实现无连接的UDP通信  原始套接字：实现IP数据包通信 |

3.流具有的三种基本操作有读取、写入、查找。

|  |
| --- |
| 第三版网络编程教材中没找到这个知识点 |

4.电子邮件由三部分组成: （信封）（首部）（正文）。

|  |
| --- |
| 对于SMTP协议C#进行了很多封装，使我们使用起来相当方便  SMTP是英文Simple Mail Transfer Protocol的缩写，意为简单邮件传输协议，默认端口为25  在SMTP协议中，电子邮件由三部分组成，信封、首部和正文。 |

5.使用组播时通过设置 Ttl 属性值设置允许路由器转发的最大数目。

|  |
| --- |
| 课本P243，使用多播时，应注意的是该对象TTL值的设置以改变允许路由器转发的最大数目。利用UdpClient对象的Ttl属性可修改TTL的默认值，例如：  UdpClient udpClient = new UdpClient( );  udpClient.Ttl = 50; |

6.将字节序列转化为字符串称为\_解码\_将字符串转化为字节序列的过程为编码\_。

|  |
| --- |
| 在网络通信中，很多情况下通信双方传达的都是字符信息。但是，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。  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。  对于Unicode字符来说，编码是指将-组 Unicode字符转换为一个字节序列的过程，解码则是将一个字节序列转换为一组 Unicode字符的过程。 |

1. 简答题

1.简述利用TcpListener类及TcpClint类编写同步TCP服务端程序的基本步骤。(6 分)

答: 1、创建一个TcpListener对象，在指定的端口进行监听。(2分)2、利用TcpListener 的AcceptTcpClient对象接收客户端的来连接请求。根据创建的TepClient对象,利用TepClient对象的GetStream方法得到NetworkStream对象，然后利用NetworkStream对象的Write方法和Read方法分别与每个连接的客户进行数据传输(2分)3、与对方通信结束后，关闭与对方的连接，释放与TcpClient 关联的所有资源。（2分）

|  |
| --- |
| 第九章TCP编程知识  这个题要注意是服务端：  TCPListener和TCPClient的编程流程    用TcpListener和TcpClient编写TCP应用程序的一般步骤  编写TCP服务端代码的一般步骤  1）创建一个TcpListener对象，然后调用该对象的Start方法在指定的端口进行监听。  2）在单独的线程中，循环调用TcpListener对象的AcceptTcpClient方法接收客户端连接请求，并根据该方法返回的结果得到与该客户端对应的TcpClient对象。  3）每得到一个新的TcpClient对象，就创建一个与该客户端对应的线程，然后通过该线程与对应的客户端通信。  4）根据传送信息的情况确定是否关闭与客户端的连接。  3．用TcpListener和TcpClient编写TCP应用程序的一般步骤  （2）编写TCP客户端代码的一般步骤  1）利用TcpClient的构造函数创建一个TcpClient对象，并利用该对象与服务端建立连接。  2）利用TcpClient对象的GetStream方法得到网络流，然后利用该网络流与服务端进行数据传输。  3）创建一个线程监听指定的端口，循环接收并处理服务端发送过来的信息。  4）完成通信工作后，向服务端发送关闭信息，并关闭与服务器的连接。 |

2.解决TCP协议的无消息边界问题一般有几种方法?它们适合哪些场合? (6分)

答: 1)发送固定长度的消息。适用于信息长度固定的场合。

2)将消息长度与消息一起发送。 此方法适用于任何场合。

3)使用特殊标记分隔消息。适用于消息本身不包含特殊标记的场合。6分(只答方法不答适用场合给4分)

|  |
| --- |
| 第九章的知识  PPT上关于此知识点的描述：  解决TCP无消息边界问题的办法  （1）发送固定长度的消息  这种办法适用于消息长度固定的场合。  （2）将消息长度与消息一起发送  一般在每次发送的消息前面用4个字节表明本次消息的长度，然后将其和消息一起发送到对方；对方接收到消息后，首先从前4个字节获取实际的消息长度，再根据消息长度值依次接收发送方发送的数据。  这种办法适用于任何场合。  （3）使用特殊标记分隔消息  使用特殊分隔符对消息进行分隔。  这种办法主要用于消息本身不包含特殊标记的场合。 |

3. P2P与传统C/s相比有何优缺点? (6 分)

答：在传统的“服务器/客户端"应用系统中，客户端与服务器有明确的分界，而使用P2P技术实现的每个计算机节点既是客户端，也是服务器。(3分)采用这种方式，不会像C/S模式那样带来服务器网络带宽瓶颈，造成网络堵塞的问题。同时，P2P技术更加符合分布式计算的理念。(2分)但是P2P技术同样带来了网络安全问题(1 分)

|  |
| --- |
| 第一章的知识  PPT上关于此知识点的描述：  C/S（Client/Server）也叫C/S模式、C/S架构或C/S模型，它是在分布式的基础上进一步抽象出来的编程模型。  C/S模式及其特点：  胖客户端应用程序编程架构，其主要工作都在客户端运行；例如QQ、飞信、360安全卫士等客户端软件。  C/S既适用于实际的应用程序，又适用于真正的计算机部署。从程序实现的角度来说，客户端和服务端实际是计算机上的两个进程的交互。服务端进程逐一等待并处理客户端请求。运行服务端进程的计算机系统一般通过所提供的服务来命名。  P2P  　　对等连接(peer-to-peer，简写为 P2P)是指两个主机在通信时并不区分哪一个是服务请求方还是服务提供方。  　　只要两个主机都运行了对等连接软件（P2P 软件），它们就可以进行平等的、对等连接通信。  　　双方都可以下载对方已经存储在硬盘中的共享文档。  　　对等连接方式的特点  　　　　对等连接方式从本质上看仍然是使用客户服务器方式，只是对等连接中的每一个主机既是客户又同时是服务器。  　　　　例如主机 C 请求 D 的服务时，C 是客户，D 是服务器。但如果 C 又同时向 F提供服务，那么 C 又同时起着服务器的作用。 |

4..NET基类库使用的私钥算法类采用的加密模式是密码本模式还是密码块链模式?如果有两个内容相同的原始数据块,使用.NET提供的对称加密类，使用相同的私钥进行加密，那么得到的加密后的两个数据块是否相同?并给出理由。(6 分)

答:

密码块

不相同

（课本P69）.NET类库中提供的块密码加密模式默认使用CBC模式。该模式通过一个秘钥Key和一个初始化向量对数据执行加密转换，并且和上一个数据库的值相关。

|  |
| --- |
| 对称加密也称为私钥加密、采用私钥算法，加密和解密数据使用的是同一个密钥。  改进后的算法是密码块链 (CBC，Cipher Block Chaining) 模式，即----使用私钥和初始化向量 (IV，Initialization Vector) 共同加密。实现思路：用“IV”和“私钥”加密第一个纯文本块，每个后续纯文本块都在加密前先与前一个密码文本块进行按位“异或”运算，从而解决了ECB模式存在的安全问题，即：虽然原始数据相同，产生的加密后的数据也不会相同。 |

5.简述POP3协议工作原理。(6分)

答:未讲，略

四、综合应用题

1.假设某主机网络地址为192.168.1.8、 22.89.116.49， 主机名为WLKF.别名为www another com。分析下面的程序，并写出程序的运行结果。

(10分)

private void buttonl\_ Click(object sender, EventAIgs e)

{

sting hostName = Sytstn.t.Dmus.CGclHoslNameO;

Console.WriteLine(hostName);

PAddress[] ips = Dns.GetHostAddresses(hostName);

foreach(IPAddress ip in ips)

{

Console.WriteLine(ip.TostringO).

}

IPHoslsEntry mc = DnsCcHostury();

foreach(string tenp in me.Aliases)  
{

Console. Writeline(temp);

}

IPAddress broadCastIp = IPAddressBroadcast;

Console.WriteL ine(broadCastIp.ToString0).

写出程序的运行结果

答:①

②

③

④

2.方法EnCryprText利用TripleDES算法对字符串的加密，下 面列出了加密方法的部分代码，选择最确切的答案，将完成事件功能的代码填写完整。(10 分)

A cryptoStream B memoryStream C ToAmay0

D ToCharAmay0 E Abort F Close0

G CreateEncryptor(Key. IV) H CreateDecryptor(Key. I)

I Write J Read

// toEncryprt待加密字节数组

//IKey算法使用密钥

//IV算法使用向量

private byte EncryptText(DYTEI||toEncrypt.byte||Key. byte||IV)

//创建一个内存流

MemoryStream memoryStream = new MemoryStream0:

//使用内存流、私钥和IV创建加密流，补充参数.

CryptoStream cryptoStream = new CryptoStream(\_\_\_①memorystream\_\_\_)

new TipleDESCrypluScrviceProvider0.②(CreateEncryptor(Key. IV))\_

CryptostreamMode.③Write\_\_);

Try{

}

cryptoStream.Write(toEncrypt, 0, toEncrypt. Length)

eryplostream.FlushFimnalBlock();

byte[] encryptedBytes = memoryStream.(④ToCharArray())

return encryptedBytes;

}

catch (CryptographicException en)

{

throw new Exception("加密出错: "+err.Message);

}

finally

{

//关闭加密流

cryploStream.(⑤Close());

memoryStream.Close().

}

}

2、(每空2分，共10分)

B或memorystream

G或CreateEncryptor(Key. IV)

I或Write

C或ToCharArray0

F或Close0

3、(每空2分，共10分)

3.现有一个Windows应用程序，该程序利用组播技术实现网络会议讨论。该会议使用组播组地址为224.0.0.1,通讯端口为3037。下面是完成此功能的核心代码框架，在下列选项中选择最确切的答案，将完成事件功能的代码填写完整。(10 分)

A.UTF8 B GetBytes Cip D iep E ref Termnote

F remote G IPAddress H GetValue I ASCII

J GetString K UdpClient L TcpClient M Close0

N Abort()

将SendToAll方法填写完整。该方法实现加入到组播组中，并向组播组发送组播消息的功能。

private void SendToAl(String strMessage)

{

1. UdpClient） myClient = new （①UdpClient）

PAddress ip = IPAddress.Parse("224.00.1);

myClient.EnableBroadcast = true;

IPEndPoint iep = new IPEndPoint(ip,3037):

myClient.JoinMuticastGroup(②ip)

myClient.Ttl = 30;

Byte[] bytes= Encoding.UTF8.③GetBytes(strMessage)

Try{

//向子网发送信息

myClient.Send(bytes,\_bytes.Length, \_④iep\_):

}

catch (Exception err)

{ MessageBox.Show(err.Message.“发送失败"):}

Finally

{

myClient.⑤Close()；

}

}

答案：

K或UdpClient

C或ip

B或GetBytes

D或iep

M或Close()

**C#网络编程第三套**

一、选择题

1.下列关于进程和线程的描述中哪个是错误的(C)

A.线程池适用于需要多个线程而实际运行时间比较短的场合。

B.在C#中可使用lock语句实现线程同步。

C.执行Thread实例的Start方法就意味着线程实例立即执行。

D.一个线程要么是前台线程要么是后台线程。

|  |
| --- |
| **线程的生命周期经历的状态：**  **新建**：**new**创建线程对象时  **就绪**：调用**start()**方法时  **运行**：调用**run()**方法时  **阻塞**： 多种原因可导致阻塞  **死亡**：多种原因 |

2.在C#中，使用哪个类完成域名解析()

A. IPAddress B. IPEndPoint C. IPHostEntry D. Dns

|  |
| --- |
| 复杂的网络应用程序中，需要掌握几个最基本的类：  1、提供网际协议IP地址的IPAddress类, 提供了对IP地址的转换和处理功能。  2、包含IP地址和端口号的IPEndPoint类；IPEndPoint是与IPAddress概念相关的一个类，它包含应用程序连接到主机上的服务所需的主机和端口信息。它由两部分组成，一个是主机IP地址，另一个是端口号。  3、为Internet或Intranet主机提供信息容器的IPHostEntry类。IPHostEntry类将一个域名系统（DNS）的主机名与一组别名和一组匹配的IP地址关联。该类一般和Dns类一起使用。  4、DNS（Domain Name System，域名系统）是因特网的一项核心服务，它可以将域名和IP地址相互转换。System.Net命名空间下的Dns类提供了方便的域名解析功能，可利用它从Internet域名系统检索指定主机的信息。  //定义远程计算机端点信息  IPEndPoint sender = new IPEndPoint(IPAddress.Any, 0）；  可知，网络通信时使用的端点是IPEndPoint |

3.关闭不带有Windows界面进程的方法为(B)

A. CloseMainWindow B. Kill C. Abort D . Join

|  |
| --- |
| C、D选项是线程管理中提供的方法，题目问的是进程。所以应该从A、B中选，A是关闭具有用户界面的进程方法，所以选B，强制终止进程。  进程终止：  Kill方法：强制终止进程   CloseMainWindow方法：关闭具有用户界面的进程   Close方法：释放与此组件关联的所有资源   HasExited属性：指示关联进程是否已终止   WaitForExit方法：设置等待关联进程退出的时间，并在该段时间结束前或该进程退出前，阻止当前线程执行。  线程管理（Thread类）   Thread类提供的常用方法   Start方法：启动线程   Join方法：将指定的线程合并到当前线程中，并阻止当前线程执行，直到指定的线程终止或经过了指定的时间为止   Sleep方法：将当前线程阻止指定的毫秒数，零(0)表示应挂起此线程以使其他等待线程能够执行   Abort方法：在调用此方法的线程上引发ThreadAbortException，以开始终止此线程的过程。调用此方法通常会终止线程。 |

4.利用FTP协议上传及下载图形图像类的文件时，为了保证数据不发生变化,选择的传送方式应为（）

A.二进制 B. ASCII C.主动模式 D.被动模式

|  |
| --- |
| A、B是传输方式，C、D是工作模式（第一套卷子里的填空题有）  FTP的文件传输方式有两种：  　1、ASCII模式。ASCII模式一般用来传输文本文件；  　2、binary（二进制）模式。除了文本文件外的其他文件用binary（二进制）模式。 |

5.下列FTP命令中不需要建立数据连接的命令为

A. LIST B. RETR C. STOR D. DELE

|  |
| --- |
| 未讲，不考 |

6. 为了确定是否是创建控件的线程访问该控件的对象，需要检查的属性是(B).

A. Visible B. InvokeRequired C. CanSelect D. Enabled

|  |
| --- |
| 默认情况下，.NET框架都不允许在一个线程中直接访问另一个线程中的控件。   为了解决死锁以及异步执行过程中的同步问题，WPF中的每个元素（包括根元素）都有一个Dispatcher属性。   要在后台线程中与用户界面交互，可以通过向WPF控件的Dispatcher注册工作项来完成。   注册工作项的常用方法有两种：Invoke方法和InvokeAsync方法。 |

7..NetFramework中UdpClient类提供的加入组播组的方法为(A)

A. JoinMuticastGroup B. DropMuticastGroup

C. InvokeRequired D. Connect

|  |
| --- |
| A 加入多播组  利用UdpClient对象的JoinMulticastGroup方法可加入到指定的多播组中，例如：  //创建UdpClient的实例并设置本地监听的端口号  UdpClient udpClient=new UdpClient(8001);  udpClient.JoinMulticastGroup(IPAddress.Parse("224.0.0.1"));  B 退出多播组  利用UdpClient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。例如：  udpClient.DropMulticastGroup(IPAddress.Parse("224.100.0.1"));  补充 是否允许接收多播  利用UdpClient对象的MulticastLoopback属性可控制是否允许接收多播信息。该属性默认为true，即允许接收多播。  C C#中禁止跨线程直接访问控件，InvokeRequired是为了解决这个问题而产生的，当一个控件的InvokeRequired属性值为真时，说明有一个创建它以外的线程想访问它。  D Connect方法  **建立默认远程主机。**因为UDP是无连接的，所以不会阻止调用该方法的线程。另外，如果打算接收多路广播数据报，不要调用Connect方法，否则从指定的默认地址以外的地址到达的任何数据报都将被丢弃。但是，如果在调用Send方法时指定了IPAddress.Broadcast，则可以将数据广播到默认的广播地址255.255.255.255 |

8.接收邮件经常使用的协议是(C)

A. TCP协议 B. UDP协议 C. POP3协议 D. SMTP协议

|  |
| --- |
| POP3协议用于接收或下载邮件，默认端口110 SMTP协议用于传输或发送邮件，默认端口25  ftp文件传输协议使得[主机](http://baike.haosou.com/doc/5331327-5566564.html)间可以共享文件，默认使用TCP的21端口标识  P2P Peer-to-peer 是一类允许一组用户互相连接并直接从用户硬盘上获取文件的网络。 |

9. 下列描述中错误的是(D)

A.对等模式 B.设计模式可分为单纯型架构和混合型架构两种

C.常见的P2P应用有电驴、BT、QQ、迅雷

D. P2P是一种新的协议

|  |
| --- |
| P2P传输协议起源较早，常见新的协议有IPFS协议等 |

10..NET中提供的用来将数据流链接到加密转换流的类为(B)

A.MemoryStream B.CryptoStream C.FileStream D.StreamReader

|  |
| --- |
| 第四章提到常见的数据流  文件流（FileStream）、内存流（MemoryStream）、网络流（NetworkStream）、加密流（CryptoStream）  StreamReader和StreamWriter类（操作文本数据时，一般用StreamReader和StreamWriter类来实现。）  BinaryReader和BinaryWriter类（System.IO命名空间还提供了BinaryReader和BinaryWriter类以二进制模式读写流，更方便于对图像文件、压缩文件等二进制数据进行操作。） |

11. 不对称算法也称什么算法(B)

A.私钥加密算法 B.公钥加密算法 C.公钥/私钥加密算法 D. Hash算法

|  |
| --- |
| 不对称加密   不对称加密也叫公钥加密。这种技术使用不同的加密密钥与解密密钥，是一种“由已知加密密钥推导出解密密钥在计算上是不可行的”密码体制。   不对称加密产生的主要原因有两个：   一是为了解决对称加密的密钥管理问题；   二是数字签名需要用不对称加密来实现。  A选项：  对称加密也称为私钥加密、采用私钥算法，加密和解密数据使用的是同一个密钥。 |

12.利用Socket实现面向连接的通信时，应使用的套接字为( )

A.流式套接字 B.数据报套接字 C.原始套接字 D.其他套接字

|  |
| --- |
| 流式套接字：实现面向连接的TCP通信  数据报套接字：实现无连接的UDP通信  原始套接字：实现IP数据包通信 |

13.下列说法正确的是()

A.向本地广播地址发送的信息可以穿越子网.

B.数字签名即实现数据身份认证

C.数字签名可实现数据身份认证和完整性验证

D. RSA算法为对称加密算法

|  |
| --- |
| A、第一套有这个选项，本地广播发送的地址不可以穿越子网。  B、除数据身份验证外还有数据完整性验证。  C、数字签名的实现原理，数字签名是利用私钥加密必须用公钥解密这个原理来实现的。在应用程序中，可以利用数字签名实现数据身份验证和数据完整性验证。数据身份验证是为了验证数据是不是持有私钥的人发送的；数据完整性验证则用于验证数据在传输过程中是否被修改过。  D、RSA算法：RSA算法不仅可用于不对称加密和解密，还能用它实现数字签名  PPT中的数字签名知识点：  通过Internet下载文件后，怎样知道下载的文件是否和原始文件完全相同呢？发送方通过Internet发送数据后，接收方如何验证接收的数据在网络传输过程中是否被修改过？   这就是数字签名的用途。   如果通信双方希望确保信息是来自对方而不是来自第三方，需要使用数字签名进行身份验证。   另外，数字签名还可以防止特定一方否认曾发送过的信息。  利用私钥加密必须用公钥解密这个原理实现。 |

14.下面关于PNRP协议错误的是()

A.对等点名称由Authority节和Classifier节两部分构成。

B.安全型的对等点Authority节的值为0。

C.群是指-组可以通过网络相互通信的节点。

D.整个IPv6因特网仅有个全局群，代表IPv6因特网上的所有对等节点。

|  |
| --- |
| 第三版网络编程未讲 |

15.SMTP命令中指定邮件正文的命令是()

A DATA B.ECLO C BEGIN D AUTH

|  |
| --- |
| 第三版网络编程未讲 |

二、填空题

请从下列选项中选择合适的内容填入对应的括号内:

Process、Thread、lock、NetWork、ForeGroundWorker、BeckGroundWorker、DoMork、Event、Result、AsyncResult、 DialogResult、IasyncResult、RSA、DSA、 ASS、 GET、 SET、 HEAD, DNS、POST、UP、DOWN、TCP、UDP、HTTP、LODA、FTP、P2P、XML、TTL、进程、线程、主动、被动、同步、异步、数据、读取、组合、写入、控制、迭代、正文、编码、解码、信封、身份、首部、组播、广播、公钥、私钥、密钥、密码、对称、完整性、参数、冲突、端口、文件、网络

1.线程(同步)是指多个线程之间存在先后执行顺序的关联关系。

|  |
| --- |
| 同步执行和异步执行   一种是执行某语句时，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。   一种是执行某语句时，不管该语句是否完成，都会继续执行其后面的语句，这种执行方式称为异步执行。 |

2. 流具有的三种基本操作有读取、写入、查找。

|  |
| --- |
| 第三版网络编程教材中没找到这个知识点 |

3.电子邮件由三部分组成: （信封）（首部）（正文）。

|  |
| --- |
| 对于SMTP协议C#进行了很多封装，使我们使用起来相当方便  SMTP是英文Simple Mail Transfer Protocol的缩写，意为简单邮件传输协议，默认端口为25  在SMTP协议中，电子邮件由三部分组成，信封、首部和正文。 |

5.使用组播时通过设置 Ttl 属性值设置允许路由器转发的最大数目。

|  |
| --- |
| 课本P243，使用多播时，应注意的是该对象TTL值的设置以改变允许路由器转发的最大数目。利用UdpClient对象的Ttl属性可修改TTL的默认值，例如：  UdpClient udpClient = new UdpClient( );  udpClient.Ttl = 50; |

5.在网络技术中，为了解决与多个应用程序同时进行通信，可以利用（端口)号来区分目标进程。

|  |
| --- |
| 在网络上，识别远程主机的信息主要由两部分组成：  一是主机标识，用于识别与本地计算机通信的是哪台远程主机；  二是端口号，用于识别和远程主机的哪个进程通信。 |

6.根据连接方向不同，FTP提供了两种文件传送方式，一种是（主动)模式，另一种是(被动)模式。

|  |
| --- |
| 根据传输文件的不同，FTP的文件传输方式有两种：  　1、ASCII模式。ASCII模式一般用来传输文本文件；  　2、binary（二进制）模式。除了文本文件外的其他文件用  根据连接方向的不同，FTP还支持两种模式：  1、一种方式叫做Standard (也就是 PORT方式，主动方式) Standard模式 FTP的客户端发送 PORT 命令到FTP服务器。  2、一种是 Passive (也就是PASV，被动方式)。 Passive模式FTP的客户端发送 PASV命令到 FTP Server。  这道题问的根据连接方向不同，所以填主动模式和被动模式 |

三、简答题

1.什么是线程同步?为什么需要使用线程同步? c#提供了什么语句明以简单地实现线程的同步? (6分)

答: 多个线程同时运行时，可能会因为线程之间的逻辑关系而而决定最先执行,谁后执行。这就是线程同步。同步是多线程中一个非常重要的概念。所谓同步，是指多个线程之间存在先后执行顺序的关联关系。使用多线程可以解决了吞吐量和响应速度的问题，但同时也带来了资源共享问题，如死锁和资源争用。如果一个线程必须在另一个线程完成某个工作后才能继续执行，则必须考虑如何让其保持同步，以确保在系统上同时运行多个线程而不会出现死锁或逻辑错误。

C#提供了lock 语句来实现线程的同步。

2.简述HTTP的特点(6分)

答: HTTP以TCP方式工作。

HTTP是无状态的。

HTTP使用元信息作为标头。

3.什么是广播、组播?两者有什么区别? (6分)

答: 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。全球广播使即点分十进制的255.255.255.255,这个特定的广播地址表明数据包的目的地是网络上的所有设备。但是由于路由器会自动过滤掉全球广播，所以使用这个地址没有实际意义。使用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。组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。

4. P2P 与传统Cs相比有何优缺点? (6 分)

答: 在传统的“服务器/客户端”应用系统中，客户端与服务器有明确的分界，而使用P2P技术实现的每个计算机节点既是客户端，也是服务器。采用这种方式，不会像C/S模式那样带来服务器网络带宽瓶颈，造成网络堵塞的问题。同时，P2P技术更加符合分布式计算的理念。其所倡导的计算能力边缘化、计算资源共享等思想，刚好与网格技术不谋而合。

5. .NET 基类库使用的私钥算法类采用的加密模式是密码木模式还是密码块链模式?如果有两个内容相同的原始数据块,使用.NIT提供的对称加密类，使用相同的私钥进行加密，那么得到的加密后的两个数据块是否相同?并给出理由。(6分)

答: 密码块、不相同

（课本P69）.NET类库中提供的块密码加密模式默认使用CBC模式。该模式通过一个秘钥Key和一个初始化向量对数据执行加密转换，并且和上一个数据库的值相关。

|  |
| --- |
| 对称加密也称为私钥加密、采用私钥算法，加密和解密数据使用的是同一个密钥。  改进后的算法是密码块链 (CBC，Cipher Block Chaining) 模式，即----使用私钥和初始化向量 (IV，Initialization Vector) 共同加密。实现思路：用“IV”和“私钥”加密第一个纯文本块，每个后续纯文本块都在加密前先与前一个密码文本块进行按位“异或”运算，从而解决了ECB模式存在的安全问题，即：虽然原始数据相同，产生的加密后的数据也不会相同。 |

四、综合应用题

1、现有一个域名析器，用户输入主机名或INS域名以后，能在列表框中显示与主机名对应的IP地址。下面已经给出了代码框架，请根据题意从备选答案中选择合适答案使程序完整。(共10分，每空2分)

Using System.Net;、Using System.Sockets;、Dns.GetHostEntry、 Dns.GetHostName、IPAddress、IPEndPoint. ipInfo.AddressList[0]. iplnfo.HostName、 ipInfo.AddressList

(1)引用命名空间

\_\_\_\_\_[①Using System.Net]\_\_

(2) button按钮事件

private void button1\_ Click(object sender, EventArgs e)

{

string hostName = this.txtHostName.Text;

listBox1.Items.Clear();

//根据主机名或域名获得IPHostEntry实例

IPHostEntry ipInfo =\_[②Dns.GetHostEntry](hostName);

//把IP地址添加到listbox1中

foreach ( [③IPAddress] ip in [④ipInfo.AddressList])

{

listBox1.Items.Add(ip.IoString());

}

//显示主机名

txtDnsName.Text= [⑤iplnfo.HostName]

}

}

2、同步TCP应用编程中，编写服务器端的应用程序和客户端应用程序有一般的步骤。(共20分)

(1)下面给出了服务器应用程序的框架，请从备选答案中选择合适的答案使程序完整。(每空2分，共14分)

IPAddress.TryParse、IPAddress.Parse、Listen（）、Start（）、Close（）、ShutDown（）、Accept（）、AcceptTcpClient（）、 AcceptSocket（）、 GetStream()、 GetBytes 、new NetWorkStream（）、GetString

class Program

{

static void Main(string[] args)

{

TcpListener server = null;

Try

{

Int port = 13000;

//将“127.0.0.1”转换成IPAddress类的实例

IPAddress localAddr = \_\_[①IPAddress.Parse]\_\_\_("127.0.0.1");

server = new TCPListener(localAddr,port);

//启动监听

server.\_ [②Start（）]\_

Byte[] bytes = new Byte[256]:

String data = null;

while (true)

{

Console.Write("Waiting for a connectin... ");

//等待客户端链接

TpClient client = server.[ ③ AcceptTcpClient（）] \_\_\_\_\_

//获得网络流对象

NetworkStream nts = client.[④GetStream()]\_\_\_

int i;

while ((i = nts .Read(bytes,0.bytes.Lengtb)) != 0){  
 //接收客户端发的消息

data = Encoding. ASCII.[⑤GetString](bytes, 0, i);

Console.WriteLine("Received: {0}", data);

}

//给客户端发送消息

data = "hello,I am server!".

Byte[] usg = EocdungASCII\_\_\_[⑥GetBytes]\_\_\_ (du).

nts. Wrile(msg,0，msg,L.engtb);

Console.Writeline("Sent: {0}", data);

client.[ ⑦Close()]

}

catch (SocktException c)

{

Console.Wwritet ine(" SockelException: (0)".e);

}

Console.Read(): .

}

}

(2)请回答使用同步TCP编写客户端程序的一般步骤。(6分)

使用同步TCP编写客户端程序的一般步骤如下:

1)创建一个包含传输过程中所采用的网络类型、数据传输类型和协

议类型的Socket对象或者TCPClient对象。

2)使用Connect方法将套接字与远程服务器地址相连。

3)与服务器进行数据传输。

4)完成工作后，向服务器发送关闭消息，并关闭与服务器的连接。

|  |
| --- |
| 第九章TCP编程知识  这个题要注意是服务端：  TCPListener和TCPClient的编程流程    用TcpListener和TcpClient编写TCP应用程序的一般步骤  编写TCP服务端代码的一般步骤  1）创建一个TcpListener对象，然后调用该对象的Start方法在指定的端口进行监听。  2）在单独的线程中，循环调用TcpListener对象的AcceptTcpClient方法接收客户端连接请求，并根据该方法返回的结果得到与该客户端对应的TcpClient对象。  3）每得到一个新的TcpClient对象，就创建一个与该客户端对应的线程，然后通过该线程与对应的客户端通信。  4）根据传送信息的情况确定是否关闭与客户端的连接。  3．用TcpListener和TcpClient编写TCP应用程序的一般步骤  （2）编写TCP客户端代码的一般步骤  1）利用TcpClient的构造函数创建一个TcpClient对象，并利用该对象与服务端建立连接。  2）利用TcpClient对象的GetStream方法得到网络流，然后利用该网络流与服务端进行数据传输。  3）创建一个线程监听指定的端口，循环接收并处理服务端发送过来的信息。  4）完成通信工作后，向服务端发送关闭信息，并关闭与服务器的连接。 |

**C#网络编程第四套**

一、选择题

1.下列关于进程和线程的描述中哪个是错误(D)

A. Process组件的Kill方法用于强制终止进程，该方法是终止没有图形化界面进程的唯一方法。

B.一个进程可能包括多个线程，也可能只包括一个线程。

C. .NET提供了可以查看远程计算机进程相关信息的方法。

D. Process组件的Name属性是进程的唯一标示。

|  |
| --- |
| A、 Kill方法：强制终止没有图形化界面进程  CloseMainWindow方法：关闭具有用户界面的进程  Close方法：释放与此组件关联的所有资源  B、进程就像火车，线程就像车厢，一个火车可以有多个车厢，当然也可能只有一个车厢（具体解释在第三章课后作业中）  C、由PPT中内容可知，可以通过GetProcess方法获取远程计算机上的所有进程：  //获取本机所有进程  Process[] myProcesses = Process.GetProcesses( );  //获取网络上远程计算机的所有进程  Process[] myProcesses = Process.GetProcesses("192.168.0.1");  D、除了Name属性外，还有ID、端口号等 |

2.下面关于前台线程和后台线程的描述中，错误的描述是(A)

A.某个进程的所有前台线程都终止后，所有属于该进程的后台线程不会立即停止，仍会继续执行。

B.创建一个Thread实例后，系统默认此实例为前台线程。

C.通过设置Thread实例的IsBackground属性，可以指定该线程是前台线程还是后台线程。

D.线程池中的线程默认都是后台线程。

|  |
| --- |
| 一个线程要么是后台线程要么是前台线程。  A、后台线程与前台线程类似，区别是后台线程不会影响进程终止。属于某个进程的所有前台线程都终止后，公共语言运行库就会结束该进程，而且所有属于该进程的后台线程也都会立即停止，而不管后台工作是否完成。  B、通过创建并启动新的Thread对象而生成的线程都是前台线程。   C、利用Thread对象的IsBackground属性，可以设置或判断一个线程是后台线程还是前台线程。通过将某个线程的IsBackground属性设置为true，使其变为后台线程。   D、默认情况下，属于托管线程池的线程（即其IsThreadPoolThread属性为true的线程）都是后台线程。 |

3.下面对于端口号说法不正确的是()

A.在网络通信中，定义逻辑意义的端口是为了解决与多个应用程序进行通信的问题。

B.逻辑端口地址的范围是十进制的0-65535.

C.默认情况下一个端口号可以同时被多个进程使用。

D. HTTP协议使用80做为默认端口。

|  |
| --- |
| 端口有两种含义：  一是指物理意义上的端口，如ADSL Modem、集线器的接口；  二是指逻辑意义上的端口，即进程标识，  在本书中的端口均指逻辑意义上的端口。端口地址用两字节二进制数来表示；1000以内的端口号大多被标准协议所占用；应用程序中可以自由使用的端口号一般都使用大于1000的值。  A、端口是为了解决与多个进程同时进行通信的问题。  B、可用端口地址的范围是十进制的0～65535。  C、端口号可以用来标识同一个主机上通信的不同应用程序，端口号+IP地址就可以组成一个套接字，用来标识一个进程  D、端口号的范围从0到65535，比如用于HTTP的80端口，用于FTP的21端口等。 |

4.下面关于XNA说法错误的是()

A. XNA 4.0中LoadContent方法用于加载各种素材资源。

B. XNA 4.0可以开发在Windows Phone 7.上运行的手机游戏。

C. XNA 4.0和Windows编程机制类似，以事件进行驱动。

D. XNA 4.0有固定周期循环和可变周期循环两种循环方式。

|  |
| --- |
| 未讲，不考 |

5.下列说法不正确的是(B)

A.组播组是开放的，可以随时加入到组播组中，也可以随时离开。

B.组播消息不能穿越子网。.

C.组播通信时不需要向组播组发起连接请求。

D.组播地址范围在224.0.0.0-239.255.255.255.

|  |
| --- |
| 由于广播是向子网中的所有计算机用户发送消息，没有目的性，不但增加了网络传输负担，而且资源消耗较高。组播的出现，较好地解决了这个问题。组播也叫多路广播。所谓组播是将消息从一台计算机发送到本网或全网内选择的计算机子集上，即发送到那些加入指定组播组的计算机上。组播组是开放的，每台计算机都可以通过程序随时加入到组播组中，也可以随时离开。   组播组是分享一个组播地址的一组设备。与IP广播类似，IP组播使用特殊的IP地址范围来表示不同的组播组。组播地址(1110开头的D类地址)是范围在224.0.0.0到239.255.255.255的D类IP地址。任何发送到组播地址的消息都会被发送到组内的所有成员设备上。组可以是永久的，也可以是临时的。大多数组播组是临时的，仅在有成员的时候才存在。用户创建一个新的组播组时只需从地址范围内选出一个地址，然后为这个地址构造一个对象，就可以开始发送消息了。  组播协议的优势：  组播协议的优势在于当需要将大量相同的数据传输到不通主机时，  1．能节省发送数据的主机的系统资源和带宽；  2．组播是有选择地复制给又要求的主机；  3. 组播可以穿越公网广泛传播，而广播则只能在局域网或专门的广播网内部传播；  4. 组播能节省网络主干的带宽； |

6.发送电子邮件要求用户提供用户名和密码的认证方式为(B)

A. SMTP B. ESMTP C. POP3 D. POP

|  |
| --- |
| 这是第二版传统编程技术，不考 |

7. 为了确定是否是创建控件的线程访问该控件的对象，需要检查的属性是(B).

A. Visible B. InvokeRequired C. CanSelect D. Enabled

|  |
| --- |
| 默认情况下，.NET框架都不允许在一个线程中直接访问另一个线程中的控件。   为了解决死锁以及异步执行过程中的同步问题，WPF中的每个元素（包括根元素）都有一个Dispatcher属性。   要在后台线程中与用户界面交互，可以通过向WPF控件的Dispatcher注册工作项来完成。   注册工作项的常用方法有两种：Invoke方法和InvokeAsync方法。 |

8.下列描述不正确的是()

A PNRP是目前唯一一种可以完成对等点发现的协议。

B. P2P的设计模式可以分为两大类:一种是单纯型P2P架构,没有专用的服务器。另一种是混合型P2P架构，即单纯型和专用服务器相结合的架构。

C. PNRP可完成对等名称的注册和解析。

D.不同对等方的对等名称可以相同。

|  |
| --- |
| PPT找不到，不考 |

9.实现FTP客户端功能的.NET封装后的类是()

A. FtpWebRequest、FtpWebResponse

B. FtpRequest、FtpResponse

C. WebRequest、WebResponse

D. Request、Response

|  |
| --- |
| FTP为第二版中传统编程技术，本学期没讲 |

10.PNRP协议中关于群的说法正确的是()

A. PNRP目前使用两种群:链接——本地群和全局群。

B.对等方可以发现和自己在同一个群中的其他对等方。

C.一个对等方只能在一个群中。

D..NET中的Cloud类可以获得群信息。

|  |
| --- |
| PPT找不到，不考 |

11. 下面关于FTP说法不正确的有()

A.通过FTP传输文件，双方至少需要建立两个TCP连接

B.FTP中的主动模式是指客户端主动向服务器发起连接请求。

C.FTP中RETR<filename>命令请求服务器将指定路径内的文件复制到客户端，即下载文件。

D.常见的FTP服务端软件有Server-U及IIS提供的FTP服务等，客户端软件有IE浏览器、CuteFp、 FlashFXP 等。

|  |
| --- |
| FTP为第二版中传统编程技术，本学期没讲 |

12.下面算法中属于对称加密算法的是(C)

A. RSA B. DSA C. TripleDES D. ECDH.

|  |
| --- |
| 第四章，常见的对称加密算法   常见的对称加密（私钥加密）算法有多种：   （1）DES和TripleDES加密算法   （2）RC2加密算法   （3）SHA-1加密算法   （4）AES加密算法 |

13.下面关于HTTP特点的说法不正确的有(A)

A. HTTP以UDP方式工作。

B. HTTP是无状态的。

C. HTTP使用元信息作为标头。

D.目前常见的HTTP版本是HTTP/1.1。

|  |
| --- |
| 第八章 HTTP的特点：  A、HTTP以TCP方式工作。  HTTP客户端首先与服务器建立TCP连接，然后客户端通过套接字发送HTTP请求，并通过套接字接收HTTP响应。  B、HTTP是无状态的  “无状态”的含义是，客户端发送一次请求后，服务器并没有存储关于该客户端的任何状态信息。即使客户端再次请求同一个对象，服务器仍会重新发送这个对象，而不管原来是否已经向该客户端发送过这个对象。  C．HTTP使用元信息作为标头  HTTP通过添加标头（Header）的方式向服务器提供本次HTTP请求的相关信息，即在主要数据前添加一部分信息，称为元信息（Metainformation）。  例如，传送的对象属于哪种类型，采用的是哪种编码等。  D、HTTP版本有0.9、1.0、1.1，HTTP是一个属于应用层的面向对象的协议，由于其简捷、快速的方式，适用于分布式超媒体信息系统。 |

14.下面哪些类不能用于实现HTTP协议(D)

A. HttpWebRequest类 B. HttpWebResponse类

C.Uri类. D. Address类

|  |
| --- |
| HTTP编程的传统实现技术（了解）  A、HttpWebRequestB、HttpWebResponse  CSDN:C#语言对HTTP协议提供了良好的支持，在.NET类库中提供了WebRequest和WebResponse类，这两个类都包含在System.Net命名空间中，利用这两个类可以实现很多高级的网络功能，本文中多线程文件下载就是利用这两个类实现的。 WebRequest和WebResponse都是抽象基类，因此在程序中不能直接作为对象使用，必须被继承，实际使用中，可根据URI参数中的URI前缀选用它们合适的子类，对于HTTP这类URI, HttpWebRequest和HttpWebResponse类可以用于处理客户程序同WEB服务器之间的HTTP通讯。 |

15.在C#中，使用那个类表示网络通信时使用的端点()

A. IPAddress B. IPEndPoint C. IPHostEntry D. Dns

|  |
| --- |
| 复杂的网络应用程序中，需要掌握几个最基本的类：  1、提供网际协议IP地址的IPAddress类, 提供了对IP地址的转换和处理功能。  2、包含IP地址和端口号的IPEndPoint类；IPEndPoint是与IPAddress概念相关的一个类，它包含应用程序连接到主机上的服务所需的主机和端口信息。它由两部分组成，一个是主机IP地址，另一个是端口号。  3、为Internet或Intranet主机提供信息容器的IPHostEntry类。IPHostEntry类将一个域名系统（DNS）的主机名与一组别名和一组匹配的IP地址关联。该类一般和Dns类一起使用。  DNS（Domain Name System，域名系统）是因特网的一项核心服务，它可以将域名和IP地址相互转换。  //定义远程计算机端点信息  IPEndPoint sender = new IPEndPoint(IPAddress.Any, 0）；  可知，网络通信时使用的端点是IPEndPoint |

二、填空题

请从下列选项中选择合适的内容填入对应的括号内:

Process、Thread、 前台线程、后台线程、主线程、lock、 NetWork 、ForeGroundWorker、BackGroundWorker. DoWork. Event、Result、 IPV6、AsyncResult、DialogResult、IAsyncResult、RSA、DSA、AES、GET、SET、DNS、POST、UP、DOWN. TCP、UDP、Ping、NetTest、 TTL、UDDI、HTTP、UDDA、FTP、P2P、XML、IPV4、TcpClient、TcpListener、UdpClient、进程、线程、同步、异步、数据、组合、控制、迭代、编码、解码、身份、组播、广播、公钥、私钥、密钥容器、文件、程序、密码、对称、完整性、参数、冲突、端口、网络

1.使用组播时通过设置(Ttl)属性值设置允许路由器转发的最大数目。

|  |
| --- |
| 课本P243，使用多播时，应注意的是该对象TTL值的设置以改变允许路由器转发的最大数目。利用UdpClient对象的Ttl属性可修改TTL的默认值，例如：  UdpClient udpClient = new UdpClient( );  udpClient.Ttl = 50; |

2.将字节序列转化为字符串称为(解码)，将字符串转化为字节序列的过程为(编码)。

|  |
| --- |
| 在网络通信中，很多情况下通信双方传达的都是字符信息。但是，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。  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。  对于Unicode字符来说，编码是指将-组 Unicode字符转换为一个字节序列的过程，解码则是将一个字节序列转换为一组 Unicode字符的过程。 |

3.程序在执行长时间运行的操作时，可能会导致界面失去响应，若需要及时响应用户界面，比较简单的办法之一就是使用工具箱中提供的(BackgroundWorker)组件。

|  |
| --- |
| PPT中说，例如：BackgroundWorker 组件、 PictureBox 控件等。目前该技术已经被淘汰。  应该不考 |

4.IP地址编制方案有两种: (IPV4)和(IPV6)。

|  |
| --- |
| IP地址  一个IP地址主要由两部分组成：一部分用于识别该地址所属的网络号，另一部分指明网络内的主机号。  网络号由Internet权力机构分配，主机号由各个网络的管理员统一分配。  （1）IPv4编址方案  由4个字节（十进制表示）组成的二进制值进行识别，中间用圆点分开，这种方法叫做点分十进制表示法。  （2）IPv6编址方案  每个IP地址有16个字节（128位二进制数），其完整格式用8段16进制表示，各段之间用冒号分隔。 |

5.异步编程有两种模式，一种采用基于(事件)的编程模式、一种采用基于(IAsyncResult)接口的编程模式。

|  |
| --- |
| PPT上说的异步编程的实现方式其实一共有四种，我估计是因为卷子是十年前的，所以当时只考了前两种（已经被淘汰），所以重点看后两种。  （1）传统的异步编程模型（APM）   使用IAsyncResult 设计模式的异步操作是通过名为 Begin\*\*\* 和 End\*\*\* 的两个方法来实现的，这两个方法分别指代开始和结束异步操作。  ◊ 例如：网络流对象的BeginRead 、EndRead 方法   异步控制复杂，目前该技术已经被淘汰。  （2）基于事件的异步编程设计模式（EAP）   该模式用事件驱动模型实现异步方法。   通常由有一个或若干个 \*\*\*Async 方法和一个对应的 \*\*\*Completed 事件。  ◊ 例如：BackgroundWorker 组件、 PictureBox 控件等。   目前该技术已经被淘汰。  （3）基于任务的异步模式（TAP）   .NET 4.0框架，异步编程建议的异步编程技术。  （4）改进的基于任务的异步模式（async、await、Task.Run和TAP） |

6. ( Ping)类可以方便测试与远程主机的是否连通。

|  |
| --- |
| PPT中没有找到 |

7. ( UdpClient)类的Connect方法向远程主机发起连接请求。

|  |
| --- |
| 第十章：同步发送和接收数据  在同步阻塞方式下，可以用UdpClient对象的Send方法向远程主机发送数据，用Receive方法接收来自远程主机的数据。 |

1.简要回答下列问题

(1)为什么使用多线程?多线程适用于什么场合?

(2)如何判断一个线程运行终止?说明思路。

答:

(1)多线程可以提高程序的并发性。(2分)在Client/Server模式下，服务器需要不断监听来自多个客户端的请求，这时，如果采用单线程机制的话，这个线程由于要不断的循环监听客户端请求，从而无暇处理其他的任务。实际上，当执行需要较长时间才能完成的连续操作时，或者等待网络或其他IO设备响应时，都可以使用多线程技术。(2分)

(2)设置bool类型字段，在线程开始前字段值为false,线程结束时设置字段为true.主线程不断检查是字段值，当值为ture时意味着线程终止。(2分) .

2.什么是TCP消息通信的无边界问题?解决TCP的无消息边界问题的方法有哪些?当发送图片到远程主机时需要采用哪种方法?

答:

TCP通信不保证一次发送的数据被一次接收到。

常用的方法有三种:

(1)发送固定长度的消息。

(2)将消息长度与消息一起发送。适用于图像传送

(3)使用特殊标记分隔消息

3.简述主动模式下FTP客户端和FTP服务器的工作流程。

答:

|  |
| --- |
| 传统编程技术，不考 |

4.简要回答下列问题

(1)说明Process类的CloseMainWindow和Kill方法的相同点和不同点。

(1)如果进程有图形用户界面，调用CloseMainWindow方法。如果进程没有用户界面，调用进程的Kill方法。Kill方法强制终止, CloseMainWindow方法有序终止,仅能终止带界面的进程。

(2) Process 类不能直接关闭和终止远程主机的进程，利用网络编程的知识，说明关闭远程主机进程的设计思路。

答: 思路1:利用WCF服务或者Web服务，提供关闭进程的远程方法。

思路2:利用TCP通信机制，新建服务端项目和客户端项目。

服务端接收到客户端的关闭进程的命令，例如: KILL 进程ID

利用Process类关闭进程后返回客户端关闭进程成功的信息。

5.数字签名有什么意义?简要回答如何实现数字签名。

答: 意义:为了验证接收者接收的文件是否和原始文件完全相同。如何实现:数字签名是利用私钥加密必须用公钥解密这个原理来实现的。实现思路是，发送方先将发送的消息使用哈希函数创建消息摘要，然后用私钥对消息摘要进行加密，以创建发送方的个人签名。接收方收到消息和签名后，使用发送方的公钥解密该签名，以恢复消息摘要，并使用发送方所用的同一哈希算法对该消息进行哈希运算。如果接收方计算的消息摘要与收到的消息摘要完全匹配，则接收方可以确保消息在传输过程中没有修改。注意，因为公钥不是保密的，所以任何人都可以验证签名。

四、综合应用题

1.分析下面的程序，写出程序运行结果。(10 分)

public class Program

{

public static void Main(String[] args)

{

Class1 cl = new Class1();

cl.MyStart0;

}

}

public class Class1

{

Thread tl, t2;

public void MyStart()

{

t1 = new Thread(Method1);

t2 = new Thread(Method2);

t1.Start();

t2.Start("abc");

Console.ReadLineO;

public void Methodl()

{

Console. WriteLine("tl");

}

public void Method2(object param)

{

t1.Join();

Console.WriteLine("t2");

Console.WriteLine(param.ToStringO);

}

运行结果:

t1 t2 abc

2.方法EnCryptText利用TripleDES算法对字符串的加密，下面列出了加密方法的部分代码，选择最确切的答案，将完成事件功能的代码填写完整。(10 分)

A cryptoStream B memoryStream C ToArray()

D ToCharArray() E Abort() F Close()

G CreateEncryptor(Key, IV) H CreateDecryptor(Key, IV)

I Write J Read

// toEncrypt待加密字节数组

// Key算法使用密钥

//IV算法使用向量

private byte[] EncryptText(BYTEQ] toEncrypt , byte[] Key, byte[]IV)

{

//创建一个内存流

MemorysStream memoryStream = new MemoryStream();

//使用内存流、私钥和IV创建加密流，补充参数。

CryptoStream cryptoStream = new CryptoStream([①memoryStream] \_)

new TipleDESCryptoServiceProvider0.[②CreateEncryptor(Key, IV)]\_

CryptoStreamMode.(③Write);

try

{

cryptoStream.Write(toEncrypt, 0, toEncrypt.Length);

cryptoStream.FlushFinalBlock0;

byte[] encryptedBytes = memoryStream.[④ToArray()]

retum encryptedBytes;

}

catch (CryptographicException err)

{

throw new Exception( "加密出错: " + er.Message);

}

finally{

cryptoStream. [⑤Close()]\_

memoryStream.Close();

}

3.现有一个Windows应用程序，利用TCP实现服务器端功能。下面列出了相应的事件及部分代码，请在下列选项中选择最确切的答案，将完成事件功能的代码填写完整。(10 分)

(1)单击启动监听，则在本机第一一个可用IP地址、端口3036.上启动TCP监听，等待客户端的连接，每当接收到一个客户端连接请求后，启动一个线程负责接收并处理该客户端发送过来的信息。

private TcpListener myListener,

private void button1\_ Click(object sender, EventArgs e)

{

String name = Dns.GetHostName();

//获取本机所有IPAddress

IPAddress[] alIIP =①[Dns] .GetHostAddresses(name);

IPAddress ip = allIP[0];

myListener = new TcpListener(ip,3036)

myListener.②[Start()]

Thread threadListenClient = new Thread(ListenClientConnet);

threadL istenClient.Start0;

}

(2) ListenClientConnect方法负责接收客户端的连接请求。

private void ListenClientConnect()

{

while (true)

{

TcpClient newClient = null;

Try{

//等待用户进入

newClient = myListener.③[AcceptTcpClient()]

}

catch

{ break;}

Thread receiveDataThread = new Thread(ReceiveData);

receiveDataThread.Start(newClient);

}

(3)负责接收数据的ReceiveData方法。

private void ReceiveData(object obj)

{

TcpClient client = (TcpClient )obj;

bool exitWhile = false;

while (exitWhile == false)

{

string receivestring = null;

try

{

//获取网络流

NetWorkStream stream=④[client.GetStream()]；

//利用StreamReader对网络流进行封装

StreamReader sr = new StreamReader(stream);

//利用sr读取数据

receiveString=\_[⑤sr.ReadLine]\_

catch

{

service. SetListBox("接收数据失败"):

}

}

答: (1) Dns (2) Start() (3) AcceptTcpClient()

(4) client. GetStream() (5) sr.ReadLineO

第一章复习题

1、数据通信模型分为三种[分散式]、[集中式]和[分布式]。

|  |
| --- |
| 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：数据和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。优点是硬件成本低、无冗余和不一致性问题，缺点是可靠性和灵活性低，反应慢。  分布式：是分散式系统和集中式系统的混合体，是目前比较完善的解决方案。 |

2、TCP/IP体系结构包括哪些层?AD

A 网络层 B 会话层 C 表示层 D 运输层

|  |
| --- |
| TCP/IP协议并不完全符合OSI的七层参考模型，采用了4层的层级结构，每一层都呼叫它的下一层所提供的网络来完成自己的需求。这4层分别为：  应用层：应用程序间沟通的层，如简单电子邮件传输（SMTP）、文件传输协议（FTP）、网络远程访问协议（Telnet）等。  传输层：在此层中，它提供了节点间的数据传送服务，如传输控制协议（TCP）、用户数据报协议（UDP）等，TCP和UDP给数据包加入传输数据并把它传输到下一层中，这一层负责传送数据，并且确定数据已被送达并接收。  网络层（网际层）：负责提供基本的数据封包传送功能，让每一块数据包都能够到达目的主机（但不检查是否被正确接收），如网际协议（IP）。  网络接口层：对实际的网络媒体的管理，定义如何使用实际网络（如Ethernet、Serial Line等）来传送数据。 |

3、以下哪个协议是UDP封装?

A FTP B HTTP C DNS D Telnet

|  |
| --- |
| DNS在进行区域传输的时候使用TCP协议，其它时候则使用UDP协议； |

4、以下哪款协议默认使用端口号80 ?

|  |
| --- |
| ftp文件传输协议使得[主机](http://baike.haosou.com/doc/5331327-5566564.html)间可以共享文件，默认使用TCP的21端口标识。  ssh 22  Secure Shell 安全外壳协议，专为远程登录会话和其他网络服务提供安全性的协议  http 80（8080） Hyper Text Transfer Protocol 超文本传输协议，用于网页浏览  DNS 53 Domain Name System 域名系统，域名解析  https 443 Hypertext Transfer Protocol Secure 超文本传输安全协议，用于安全浏览网页  www代理服务 8080 Apache Tomcat web server，进行网页浏览  smtp 25 Simple Mail Transfer Protocol 简单邮件传输协议  telnet 23 不安全的文本传送  pop3 110 Post Office Protocol |

5、Window的cmd命令行使用[netstat -ano]命令来查看网络配置的详细参数比如各个网卡的IP地址、MAC地址、DNS服务器列表。

|  |
| --- |
| 查看本机开放的所有端口.点开始,运行,输入cmd ,回车.输入 netstat -ano 回车.可以查看本机开放的全部端口.其中Port是端口类型,其中Local Address代表本机IP地址和打开的端口号,Foreign Address是远程计算机IP地址和端口号，State表明当前的连接状态,PID显示的是对应进程 的PID.在任务管理器 的进程中可以根据对应的PID可以找到对应的进程的。 |

6、C#中可用于网卡信息检测相关类是（AC）

A NetworkInterface类 B IPGlobalProperties类

C IPInterfaceProperties类 D IPAddress类

|  |
| --- |
| 1、NetworkInterface类  位于System.Net.NetworkInformation命名空间下；  可以检测本机有多少个网络适配器、哪些网络连接可用，并可获取某个网络适配器的型号、MAC(Media Acess Control，介质访问控制)地址和速度等信息。获取网络适配器相关信息时，要先构造NetworkInterface对象。注意不能直接使用new关键字构造该类的实例，而用该类提供的静态GetAllNetworkInterfaces方法得到NetworkInterface类型数组。每个网络适配器，都包含一个NetworkInterface对象与之对应。  2、IPInterfaceProperties类  IPInterfaceProperties类提供了检测IPv4和IPv6的网络适配器地址信息，利用该类可检测本机所有网络适配器支持的各种地址，如DNS服务器的IP地址、网关地址以及多路广播地址等。IPInterfaceProperties是一个抽象类，不能直接创建该类的实例，而是通过调用NetworkInterface对象GetIPProperties方法得到该类的实例。 |