



红磨坊后的白桦树

< 2021年6月 >

日	一	二	三	四	五	六
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

昵称: [红磨坊后的白桦树](#)
园龄: [3年11个月](#)
粉丝: [4](#)
关注: [67](#)
[+加关注](#)

搜索

- 常用链接
- [我的随笔](#)
 - [我的评论](#)
 - [我的参与](#)
 - [最新评论](#)
 - [我的标签](#)

- 我的标签
- [c#\(88\)](#)
 - [C# 基础\(41\)](#)
 - [学习\(30\)](#)
 - [ORACLE\(26\)](#)
 - [WinFrom\(21\)](#)
 - [c# 线程\(20\)](#)
 - [JS使用\(18\)](#)
 - [DataTable使用\(16\)](#)
 - [数据连接\(13\)](#)
 - [日常问题积累\(12\)](#)
 - [更多](#)

- 随笔档案
- [2021年3月\(1\)](#)
 - [2020年10月\(1\)](#)
 - [2020年9月\(2\)](#)
 - [2020年8月\(9\)](#)
 - [2020年7月\(7\)](#)
 - [2020年5月\(8\)](#)
 - [2020年4月\(10\)](#)
 - [2020年3月\(1\)](#)
 - [2020年2月\(1\)](#)
 - [2020年1月\(7\)](#)

Socket传输大文件(发送与接收)

下载

Client



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Net;
using System.Net.Sockets;
using System.Threading;
using System.IO;
using System.Windows.Forms;

namespace SocketClient
{
    class Program
    {
        [STAThread]
        static void Main(string[] args)
        {
            SocketClient();
        }

        public static string serverIp = "127.0.0.1";//设置服务端IP
        public static int serverPort = 8888;//服务端端口
        public static Socket socketClient;//定义socket
        public static Thread threadClient;//定义线程
        public static byte[] result = new byte[1024];//定义缓存
        public static void SocketClient()
        {
            socketClient = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
            IPAddress ip = IPAddress.Parse(serverIp);//获取服务器IP地址
            IPEndPoint point = new IPEndPoint(ip, serverPort);//获取端口
            try
            {
                socketClient.Connect(point);//链接服务器IP与端口
                Console.WriteLine("连接服务器中.....");
            }
            catch (Exception)
            {
                Console.WriteLine("与服务器链接失败!!!");
                return;
            }
            Console.WriteLine("与服务器链接成功!!!");
            try
            {
                Thread.Sleep(1000); //等待1秒钟
                //通过 socketClient 向服务器发送数据
            }
        }
    }
}
```

[2019年12月\(38\)](#)
[2019年11月\(10\)](#)
[2019年10月\(65\)](#)
[2019年9月\(17\)](#)
[2019年8月\(26\)](#)
[更多](#)

最新评论

[1. Re:C#并发编程经典实例\(完整版\)](#)

请问第二版有吗

--惠山脚下小学僧

[2. Re:DotNetSpeech----文本转wave语音文件](#)

你好。我想请问下你如何优化朗读完成如何释放内存呢。

--郭达·斯坦森

[3. Re:DotNetSpeech----文本转wave语音文件](#)

网上看来篇帖子。生成文件的，可是怎么都不成功

--郭达·斯坦森

[4. Re:DotNetSpeech----文本转wave语音文件](#)

@红磨坊后的白桦树 我看到百度网盘地址了。...

--郭达·斯坦森

[5. Re:DotNetSpeech----文本转wave语音文件](#)

@红磨坊后的白桦树 大佬兄弟明白。这个文件是生成的呀。我没有在我下载目录找到呢。...

--郭达·斯坦森

阅读排行榜

- [1. SQL之to_date\(\)以及关于日期处理的详解\(18778\)](#)
- [2. visual studio 2017 报错 无法下载安装文件。请检查Internet连接。然后重试\(14175\)](#)
- [3. oracle 修改字符集 为ZHS16GBK\(6218\)](#)
- [4. c# 多张图片合成一张图片\(3467\)](#)
- [5. 查看电脑使用记录（包括打开过的文件）\(3078\)](#)

评论排行榜

- [1. DotNetSpeech----文本转wave语音文件\(6\)](#)
- [2. C#并发编程经典实例\(完整版\)\(1\)](#)
- [3. 移动MAS短信平台发送短信\(1\)](#)

推荐排行榜

- [1. 清理c盘垃圾\(1\)](#)
- [2. Interlocked.Increment\(\)函数详解\(转载\)\(1\)](#)

```
string sendMessage = "已成功接到SocketClient发送的消息";//发送到服务端的
byte[] send = Encoding.UTF8.GetBytes(sendMessage);//Encoding.UTF8.G
byte[] SendMsg = new byte[send.Length + 1];//定义新的字节数组
SendMsg[0] = 0;//将数组第一位设置为0，来表示发送的是消息数据
Buffer.BlockCopy(send, 0, SendMsg, 1, send.Length);//偏移复制字节数组
socketClient.Send(SendMsg); //将接受成功的消息返回给SocketServer服务器
Console.WriteLine("发送完毕：{0}", sendMessage);

}

catch
{
    socketClient.Shutdown(SocketShutdown.Both);//禁止Socket上的发送和接受
    socketClient.Close();//关闭Socket并释放资源
}

//打开文件
OpenFileDialog ofd = new OpenFileDialog();
ofd.Title = "选择要传的文件";
ofd.InitialDirectory = @"E:\IVS\Down";
//ofd.Filter = "文本文件|*.txt|图片文件|*.jpg|视频文件|*.avi|所有文件|*.*";
ofd.ShowDialog();
//得到选择文件的路径
string filePath = ofd.FileName;//获取文件的完整路径
Console.WriteLine("发送的文件路径为：" + filePath);
using (FileStream fsRead = new FileStream(filePath, FileMode.OpenOrCrea
{
    //1. 第一步：发送一个文件，表示文件名和长度，让客户端知道后续要接收几个包来重新:
    string fileName = Path.GetFileName(filePath);
    Console.WriteLine("发送的文件名是：" + fileName);
    long fileLength = fsRead.Length;//文件长度
    Console.WriteLine("发送的文件长度为：" + fileLength);
    string totalMsg = string.Format("{0}-{1}", fileName, fileLength);
    byte[] buffer = Encoding.UTF8.GetBytes(totalMsg);
    byte[] newBuffer = new byte[buffer.Length + 1];
    newBuffer[0] = 2;
    Buffer.BlockCopy(buffer, 0, newBuffer, 1, buffer.Length);
    socketClient.Send(newBuffer);//发送文件前，将文件名和长度发过去
    //2第二步：每次发送一个1MB的包，如果文件较大，则会拆分为多个包
    byte[] Filebuffer = new byte[1024 * 1024 * 5];//定义1MB的缓存空间
    int readLength = 0; //定义读取的长度
    bool firstRead = true;
    long sentFileLength = 0;//定义发送的长度
    while ((readLength = fsRead.Read(buffer, 0, buffer.Length)) > 0 &&
    {
        sentFileLength += readLength;
        //第一次发送的字节流上加个前缀1
        if (firstRead)
        {
            byte[] firstBuffer = new byte[readLength + 1];
            firstBuffer[0] = 1;//标记1，代表为文件
            Buffer.BlockCopy(buffer, 0, firstBuffer, 1, readLength);
            socketClient.Send(firstBuffer, 0, readLength + 1, SocketFla
            Console.WriteLine("第一次读取数据成功，在前面添加一个标记");
            firstRead = false;
            continue;
        }
        socketClient.Send(buffer, 0, readLength, SocketFlags.None);
        Console.WriteLine("{0}：已发送数据：{1}/{2}", socketClient.Remote
    }
    fsRead.Close();
    Console.WriteLine("发送完成");
}
Console.ReadLine();
}
}
```



Server



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Net;
using System.Net.Sockets;
using System.Threading;
using System.IO;
using System.Windows.Forms;

namespace SocketServer
{
    class Program
    {
        [STAThread]
        static void Main(string[] args)
        {
            SocketServer();
        }

        public static string serverIp = "127.0.0.1"; //设置服务端IP
        public static int serverPort = 8888; //服务端端口
        public static Socket socketServer; //定义socket
        public static Thread threadWatch; //定义线程
        public static byte[] result = new byte[1024 * 1024 * 2]; //定义缓存
        //public static string fileName; //获取文件名
        public static string filePath = ""; //存储保存文件的路径
        public static void SocketServer()
        {
            socketServer = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
            IPAddress ip = IPAddress.Parse(serverIp); //获取服务器IP地址
            IPEndPoint point = new IPEndPoint(ip, serverPort); //获取端口
            try
            {
                socketServer.Bind(point); //绑定IP地址及端口
            }
            catch (Exception ex)
            {
                Console.WriteLine("绑定IP时出现异常: " + ex.Message);
                return;
            }
            socketServer.Listen(100); //开启监听并设定最多10个排队连接请求
            threadWatch = new Thread(WatchConnect); //创建一个监听进程
            threadWatch.IsBackground = true; //后台启动
            threadWatch.Start(); //运行
            Console.WriteLine("服务器(0)监听启动成功!", socketServer.LocalEndPoint.ToString());
            Console.ReadLine();
        }

        public static void WatchConnect()
        {
            while (true)
            {
                Socket watchConnect = socketServer.Accept(); //接收连接并返回一个新的Socket
                watchConnect.Send(Encoding.UTF8.GetBytes("服务器连接成功")); //在客户端发送消息
                Thread threadwhat = new Thread(ReceiveMsg); //创建一个接受信息的进程
                threadwhat.IsBackground = true; //后台启动
                threadwhat.Start(watchConnect); //有传入参数的线程
            }
        }

        public static DateTime GetTime()
        {
            DateTime now = new DateTime();
            now = DateTime.Now;
            return now;
        }
    }
}
```

```

public static void ReceiveMsg(object watchConnect)
{
    Socket socketServer = watchConnect as Socket;
    long fileLength = 0; //文件长度
    string recStr = null; //文件名
    while (true)
    {
        int firstRcv = 0;
        byte[] buffer = new byte[1024 * 1024 * 5];
        try
        {
            //获取接受数据的长度, 存入内存缓冲区, 返回一个字节数组的长度
            if (socketServer != null) firstRcv = socketServer.Receive(buffer);
            if (firstRcv > 0) //大于0, 说明有东西传过来
            {
                if (buffer[0] == 0) //0对应文字信息
                {
                    string recMsg = Encoding.UTF8.GetString(buffer, 1, firstRcv);
                    Console.WriteLine("客户端接收到信息: " + socketServer.LocalEndPoint.ToString());
                }

                if (buffer[0] == 1) //1对应文件信息
                {
                    SaveFileDialog sfDialog = new SaveFileDialog(); //创建SaveFileDialog
                    string spath = @"E:\存放"; //制定存储路径
                    string savePath = Path.Combine(spath, recStr); //获取存储路径
                    int rec = 0;
                    long recFileLength = 0;
                    bool firstWrite = true;
                    using (FileStream fs = new FileStream(savePath, FileMode.Append, FileAccess.Write))
                    {
                        while (recFileLength < fileLength)
                        {
                            if (firstWrite)
                            {
                                fs.Write(buffer, 1, firstRcv - 1);
                                fs.Flush();
                                recFileLength += firstRcv - 1;
                                firstWrite = false;
                            }
                            else
                            {
                                rec = socketServer.Receive(buffer);
                                fs.Write(buffer, 0, rec);
                                fs.Flush();
                                recFileLength += rec;
                            }
                        }
                        Console.WriteLine("{0}: 已接收数据: {1}/{2}", socketServer.LocalEndPoint.ToString(), recFileLength, fileLength);
                        fs.Close();
                    }
                    Console.WriteLine("保存成功!!!");
                }
                if (buffer[0] == 2) //2对应文件名字和长度
                {
                    string fileNameWithLength = Encoding.UTF8.GetString(buffer, 1, firstRcv);
                    recStr = fileNameWithLength.Split('-').First(); //获取文件名
                    Console.WriteLine("接收到的文件名为: " + recStr);
                    fileLength = Convert.ToInt64(fileNameWithLength.Split('-').Last());
                    Console.WriteLine("接收到的文件长度为: " + fileLength);
                }
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine("系统异常..." + ex.Message);
            break;
        }
    }
}

```

<https://www.cnblogs.com/macT/p/11661735.html>

```
//          SaveFileDialog sfd = new SaveFileDialog();
//          sfd.Title = "保存文件";
//          sfd.InitialDirectory = @"C:\Users\Administrator\Desktop";
//          //sfd.Filter = "文本文件|*.txt|图片文件|*.jpg|视频文件|*.avi|
//          //如果没有选择保存文件路径就一直打开保存框
//          SaveFileDialog save = new SaveFileDialog();//创建SaveFileDialog
//          string spath = @"C:\Users\admin\Desktop";//制定存储路径
//          filePath = Path.Combine(spath, fileName);//获取存储路径及文件名称
//          }
//      }
//}
//catch { }
```

```
//}
```

```
//try
//{
//    if (receiveName[0] == 0)//判断数组第一个值，如果为0则说明传的是信息
//    {
//        fileName = Encoding.UTF8.GetString(result, 1, num - 1);//提取5
//        Console.WriteLine("接收客户端的消息: {0}", fileName);
//    }
//    if (receiveName[0] == 1)//判断数组第一个值，如果为1则说明传的是文件
//    {
//        SaveFileDialog save = new SaveFileDialog();//创建SaveFileDialog
//        string spath = @"C:\Users\admin\Desktop";//制定存储路径
//        string fullPath = Path.Combine(spath, fileName);//获取存储路径及文件名称
//        FileStream fileSave = new FileStream(fullPath, FileMode.Create, FileAccess.Write);
//        fileSave.Write(result, 1, num - 1);//将数据写入到文件中
//        fileSave.Close();
//        Console.WriteLine("保存成功!!!");
//        //*****
//        while (true)
//        {
//            byte[] buffer = new byte[1024 * 1024];
//            int r = socketClient.Receive(buffer);
//            Console.WriteLine("从客户端接收到的字节数: "+r);
//            //string leng = Encoding.UTF8.GetString(buffer, 1, r - 1);
//            int receive = r - 1;
//            Console.WriteLine(leng);
//            long receive = int.Parse(leng);
//            //long receive = Convert.ToInt64(Encoding.UTF8.GetString(buffer, 1, receive));
//            Console.WriteLine("总接受字节数: " + receive);
//            long length = receive;
//            Console.WriteLine("*****");
//            if (length > 0)
//            {
//                using (FileStream fsWrite = new FileStream(fullPath, FileMode.Append, FileAccess.Write))
//                {
//                    Console.WriteLine("-----*****-----");
//                    fsWrite.Write(buffer, 1, r-1);
//                    length -= r; //减去每次保存的字节数
//                    Console.WriteLine("剩余接受字节数: "+length);
//                    Console.WriteLine(string.Format("{0}: 已接收: {1}/
```

[刷新评论](#) [刷新页面](#) [返回顶部](#)

登录后才能查看或发表评论，立即 [登录](#) 或者 [逛逛](#) 博客园首页

[【推荐】资料合集 | HarmonyOS从入门到大神学习资料下载合集](#)

[【推荐】大型组态、工控、仿真、CAD\GIS 50万行VC++源码免费下载!](#)

[【推荐】阿里云爆品销量榜单出炉，精选爆款产品低至0.55折](#)

[【推荐】限时秒杀！国云大数据魔镜，企业级云分析平台](#)

园子动态：

- [致园友们的一封检讨书：都是我们的错](#)
- [数据库实例 CPU 100% 引发全站故障](#)
- [发起一个开源项目：博客引擎 fluss](#)

最新新闻：

- [科学家打造可自我维持的绿色神经形态传感器](#)
 - [苹果要求从创作者应用Fanhouse抽成30% 否则8月下架](#)
 - [C12 Quantum Electronics完成1000万美元种子轮融资 加速量子技术发展](#)
 - [腾讯申请多个机器人商标 含此前的公开的轮腿式机器人](#)
 - [腾讯10年FPS大作《CFHD》今日全面开放：虚幻引擎打造](#)
- » [更多新闻...](#)

历史上的今天：

2019-10-12 [Web 服务器被配置为不列出此目录的内容](#)

Copyright © 2021 红磨坊后的白桦树

Powered by .NET 5.0 on Kubernetes Powered by: 博客园