**小型PACS测试系统开发说明文档v0.1**

[1.概述 - 2 -](#_Toc363825312)

[2.开发环境 - 2 -](#_Toc363825313)

[3.传输测试 - 2 -](#_Toc363825314)

[3.1界面说明 - 2 -](#_Toc363825315)

[3.2 文件说明 - 3 -](#_Toc363825316)

[3.3 实现说明 - 4 -](#_Toc363825317)

[4.压缩测试 - 6 -](#_Toc363825318)

[4.1界面说明 - 6 -](#_Toc363825319)

[4.2 文件说明 - 7 -](#_Toc363825320)

[4.3 实现说明 - 7 -](#_Toc363825321)

[5.并发测试 - 11 -](#_Toc363825322)

[5.1 界面说明 - 11 -](#_Toc363825323)

[5.2 文件说明 - 12 -](#_Toc363825324)

[5.3 实现说明 - 12 -](#_Toc363825325)

[6.服务端 - 14 -](#_Toc363825326)

[6.1界面说明 - 14 -](#_Toc363825327)

[6.2 文件说明 - 15 -](#_Toc363825328)

[6.3 实现说明 - 16 -](#_Toc363825329)

[7.附录 - 17 -](#_Toc363825330)

[7.1 使用说明 - 17 -](#_Toc363825331)

[7.2 DICOM文件样例 - 20 -](#_Toc363825332)

[7.3 DCMTK开源库 - 20 -](#_Toc363825333)

# 1.概述

本程序demo主要用于模拟小型PACS检测系统进行DICOM标准兼容性的测试，一共分为四个部分：传输测试，压缩测试，并发测试，服务端。

传输测试是检测小型PACS系统在存储医学影像过程对于DICOM标准的支持，以及在传输过程中的准确性。压缩测试是检测小型PACS系统对于医学影像压缩过程中影像的变化以及检测相关参数。并发测试是模拟多个客户端同时向PACS系统发出数据请求时系统传输效率和负荷率。

# 2.开发环境

（1）IDE：Visual Studio 2008

（2）语言：Visual C++

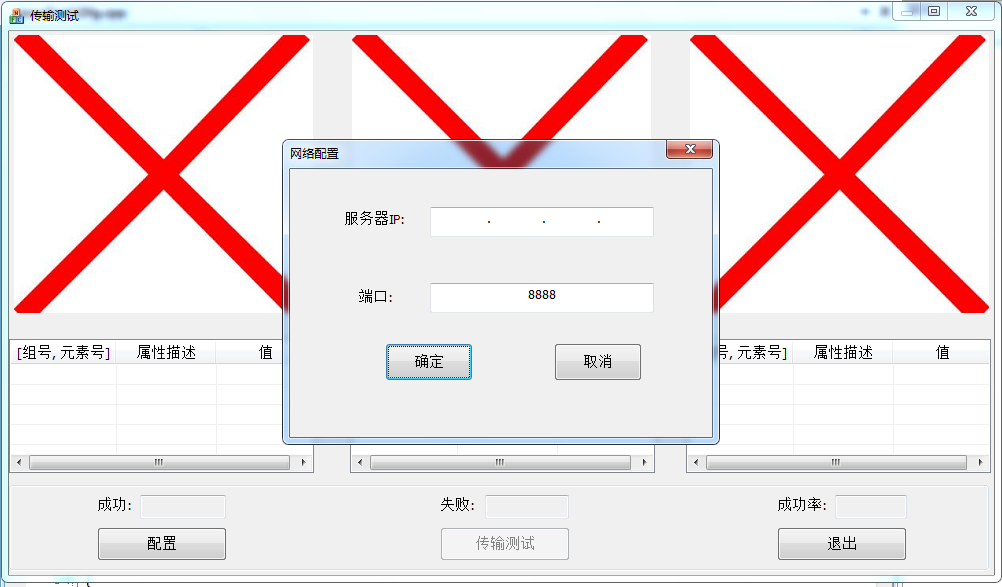
（3）系统平台：Windows 7

（4）开源库：DCMTK 3.5.4

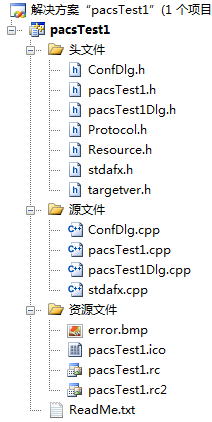
# 3.传输测试

## 3.1界面说明

程序主界面主要是显示客户端接收到的以三个为一组的DICOM文件以及三个操作按钮：配置，传输测试，退出。首先在做传输测试之前要进行网络配置，需要点击“配置”按钮，弹出网络配置子对话框（默认通信端口为8888），确认后再点击“传输测试”按钮进行DICOM文件传输测试。显示图片用picture control控件，显示信息用list control控件。如下图所示。



## 3.2 文件说明

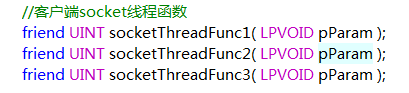
 项目文件总共分为主对话框头文件和实现文件，网络配置子对话框头文件和实现文件，以及协议头文件等。如下右图所示：

其中，ConfDlg.h和ConfDlg.cpp分别为网络配置子对话框类CConfDlg的头文件和实现文件；pacsTest1.h和pacsTest1.cpp为MFC应用头文件和实现文件；pacsTest1Dlg.h和pacsTest1Dlg.cpp分别为主对话框的头文件和实现文件，程序主要实现代码均在pacsTest1Dlg.cpp中编写；Protocol.h为协议头文件，定义了客户端和服务端通信的一些约定；资源文件中error.bmp为显示错误的位图；剩余的Resource.h，stdafx.h，stdafx.cpp和targetver.h均为MFC自动生成文件。

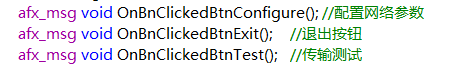
## 3.3 实现说明

每次传输测试从服务端以阻塞式socket方式传输三个DICOM文件，为每个传输单独创建一个线程进行文件接收，图像显示和信息读取等操作。同时为了避免线程间的竞争条件，加入CMutex互斥锁保证对同一文件和数据的读写正常。主要实现函数及代码如下：

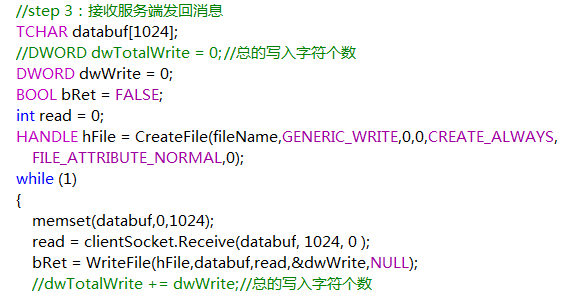
（1）Socket线程函数：



（2）按钮响应函数：



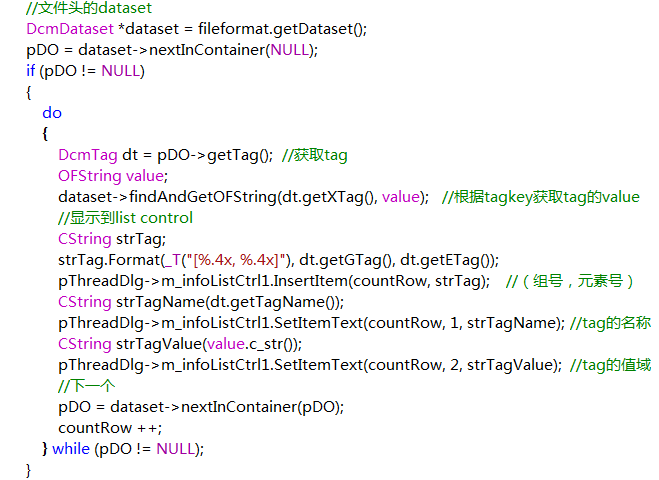
（3）文件接收：



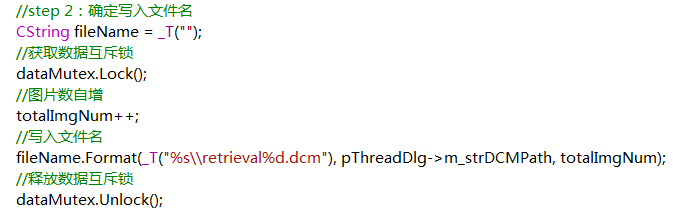
（4）图像显示：



（5）信息读取：



（6）互斥锁的使用：



另外，传输准确率的计算基于传输准确率和文件格式是否兼容DICOM 3.0两方面来衡量，即只有满足两个方面才被算作传输成功，传输不成功将显示error图片。

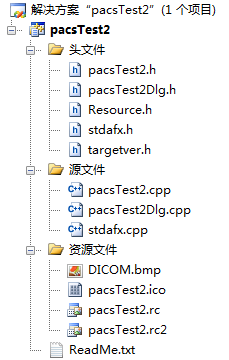
# 4.压缩测试

## 4.1界面说明

程序主界面主要是用于显示压缩前文件和压缩后文件的两个picture control，显示压缩参数的列表控件，以及四个操作按钮：选择文件，JPEG无损压缩，JPEG有损压缩，RLE无损压缩。首先在做压缩测试之前要点击“选择文件”按钮选择待压缩的DICOM文件，随后点选不同按钮就行不同压缩，同时图片控件显示压缩后图像，列表控件显示压缩过程中的参数值。如下图所示。



## 4.2 文件说明

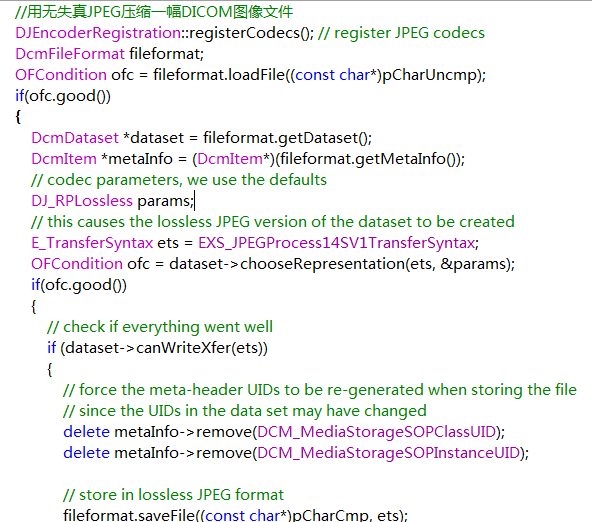
项目文件总共分为主对话框头文件和实现文件等。如右图所示。

其中，pacsTest2.h和pacsTest2.cpp分别为MFC应用头文件和实现文件；pacsTest2Dlg.h和pacsTest2Dlg.cpp分别为主对话框类CpacsTest2Dlg类的头文件和实现文件，程序主要实现代码均在CpacsTest2Dlg.cpp中编写；剩余的Resource.h，stdafx.h，stdafx.cpp和targetver.h均为MFC自动生成文件。另外资源文件中DICOM.bmp为图片控件的背景位图。

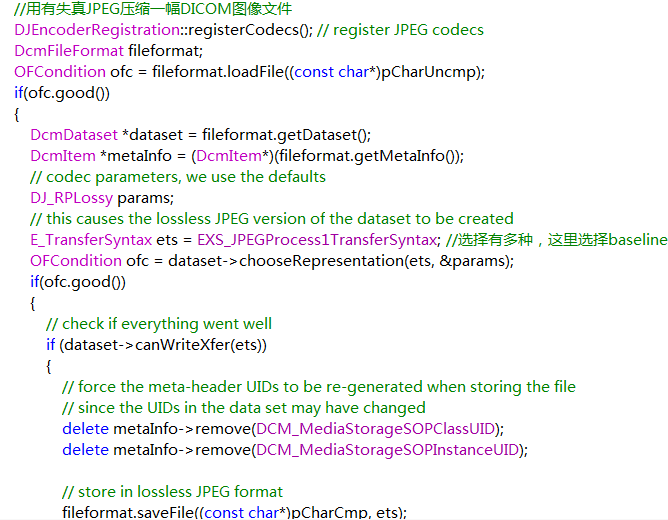
## 4.3 实现说明

每次点击“选择文件”按钮时，程序新创建文件对话框，指定DICOM文件后（符合DICOM3.0标准的文件），图片控件显示图像，随后分别点选不同压缩方式产生不同的图像效果以及压缩参数。这里的DICOM图像压缩主要借助DCMTK的类库来完成。

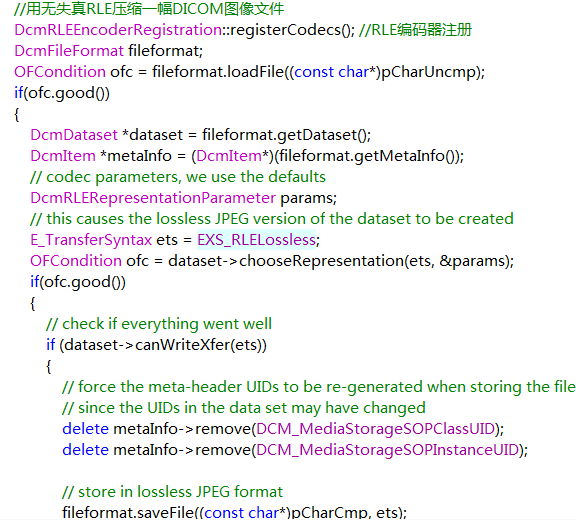
（1）JPEG无损压缩：利用DJEncoderRegistration类的静态方法registerCodecs()注册JPEG编码器，再设置编码参数为DJ\_RPLossless类的对象，再根据编码参数和压缩方式选择适应的传输语义（可有多种选择，这里选择的是EXS\_JPEGProcess14SV1TransferSyntax），随后使用DcmDataset类的chooseRepresentation()方法固定搭配好压缩方式和传输语义，最后利用DcmFileFormat类的saveFile()方法写入压缩后的文件。主要实现函数及代码如下：



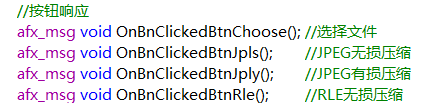
（2）JPEG有损压缩：利用DJEncoderRegistration类的静态方法registerCodecs()注册JPEG编码器，再设置编码参数为DJ\_RPLossy类的对象，再根据编码参数和压缩方式选择适应的传输语义（可有多种选择，这里选择的是EXS\_JPEGProcess1TransferSyntax，即Baseline压缩方式），随后使用DcmDataset类的chooseRepresentation()方法固定搭配好压缩方式和传输语义，最后利用DcmFileFormat类的saveFile()方法写入压缩后的文件。主要实现函数及代码如下：



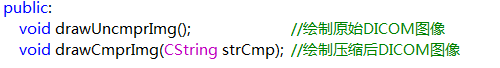
（3）RLE无损压缩：利用DcmRLEEncoderRegistration类的静态方法registerCodecs()注册RLE编码器，再设置编码参数为DcmRLERepresentationParameter类的对象，再根据编码参数和压缩方式选择适应的传输语义（仅有一种选择即EXS\_RLELossless），随后使用DcmDataset类的chooseRepresentation()方法固定搭配好压缩方式和传输语义，最后利用DcmFileFormat类的saveFile()方法写入压缩后的文件。主要实现函数及代码如下：



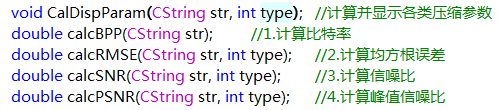
（4）按钮响应函数：



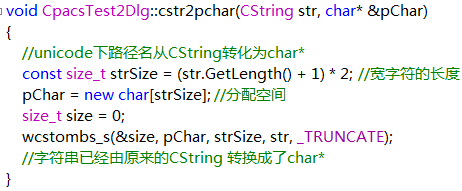
（5）绘图函数：



（6）计算压缩参数：



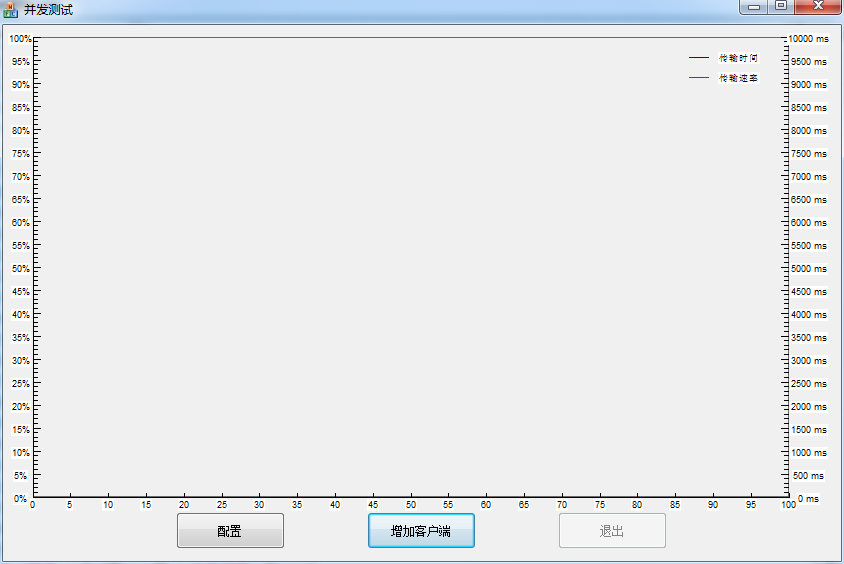
（7）Unicode下CString转换const char\*类型：由于项目字符集为Unicode，DCMTK中存在很多需要使用const char\*的地方，所以编写一个用于转换CString的方法，主要代码如下：



# 5.并发测试

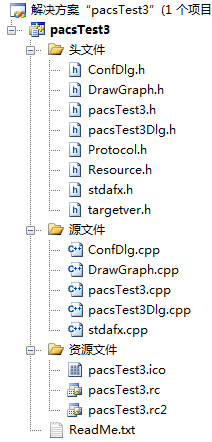
## 5.1 界面说明

程序主界面主要是用于绘制曲线的picture control，以及三个操作按钮：配置，增加客户端，退出。首先在做并发测试之前要进行网络配置，需要点击“配置”按钮，弹出网络配置子对话框（默认通信端口为8888），确认后再点击“增加客户端”按钮进行模拟多客户端请求并发测试。如下图所示。



其中蓝色曲线为传输时间曲线，红色曲线为传输速率曲线，左边坐标轴为传输速率（最高100%与1MB/s对应），右边坐标轴为传输时间（最高为10000ms，超过仍按照10000ms计算），横坐标为客户端数目。

## 5.2 文件说明

项目文件总共分为主对话框头文件和实现文件，网络配置子对话框头文件和实现文件，绘图类头文件及实现文件，以及协议头文件等。如下右图所示：

其中，ConfDlg.h和ConfDlg.cpp分别为网络配置子对话框类CConfDlg的头文件和实现文件；pacsTest3.h和pacsTest3.cpp分别为MFC应用头文件和实现文件；pacsTest3Dlg.h和pacsTest3Dlg.cpp分别为主对话框类CpacsTest3Dlg类的头文件和实现文件，程序主要实现代码均在pacsTest3Dlg.cpp中编写；DrawGraph.h和DrawGraph.cpp分别为绘图类DrawGraph的头文件和实现文件；Protocol.h为协议头文件，定义了客户端和服务端通信的一些约定；剩余的Resource.h，stdafx.h，stdafx.cpp和targetver.h均为MFC自动生成文件。

## 5.3 实现说明

每次点击“增加客户端”按钮时，程序新创建5个线程模拟5个新的客户端（最大客户端数目为100个）向服务端发起数据请求，传输双方以阻塞式socket方式传输DICOM文件。每个“客户端”传输完毕时计算前面已完成传输的“客户端”的平均传输速率和传输时间，并以结构体（GraphData）形式存入一个传输数据队列（list类）。新建一个绘图线程，用于在传输过程中利用绘图类（DrawGraph）对象从传输数据队列中取出数据进行绘图。同时为了避免线程间的竞争条件，加入CMutex互斥锁保证对同一文件和数据的读写正常。主要实现函数及代码如下：

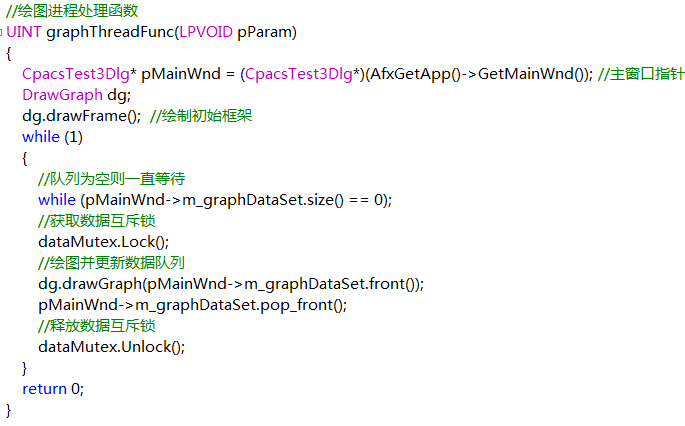
（1）Socket线程函数：



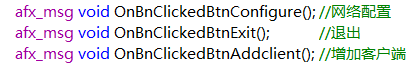
（2）绘图线程函数：



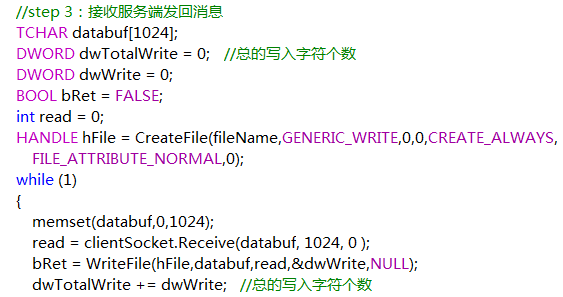
在绘图线程函数中，绘图类对象通过drawFrame()方法绘制坐标轴及刻度，得到传输数据后通过drawGraph()绘制传输曲线，drawGraph()的参数即为从队列中取出来的传输数据结构体。



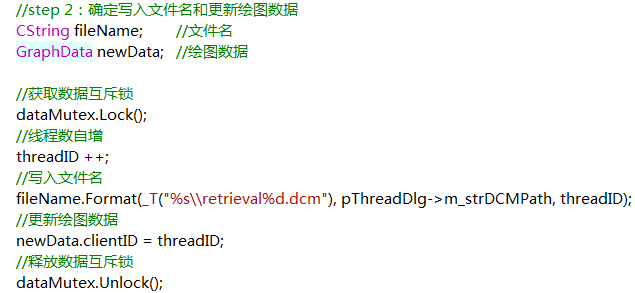
（3）按钮响应函数：



（4）文件接收：



（5）互斥锁的使用：



# 6.服务端

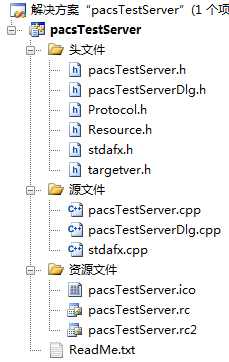
## 6.1界面说明

程序主界面主要是启动按钮，网络配置选项，以及消息显示列表控件。传输测试和并发测试之前均需要开启服务端，其中的IP地址自动获取本机IP，端口号默认8888，列表控件用于显示消息。如下图所示。



## 6.2 文件说明

项目文件总共分为主对话框头文件和实现文件，以及协议头文件等。如下右图所示：

其中，pacsTestServer.h和pacsTestServer.cpp分别为MFC应用头文件和实现文件；pacsTestServerDlg.h和pacsTestServerDlg.cpp分别为主对话框类CpacsTestServerDlg类的头文件和实现文件，程序主要实现代码均在CpacsTestServerDlg.cpp中编写；Protocol.h为协议头文件，定义了客户端和服务端通信的一些约定；剩余的Resource.h，stdafx.h，stdafx.cpp和targetver.h均为MFC自动生成文件。

## 6.3 实现说明

点击“启动服务器”按钮时，程序创建服务器线程（守护线程）用于循环监听端口并接受客户端连接。当客户端socket被服务端accept之后，服务器线程新创建线程用于处理该socket连接：接收客户端发送的请求（定义为C-MOVE消息），以阻塞式socket发回DICOM文件。同时为了避免线程间的竞争条件，加入CMutex互斥锁保证对同一文件和数据的读写正常。主要实现函数及代码如下：

（1）服务器线程函数：



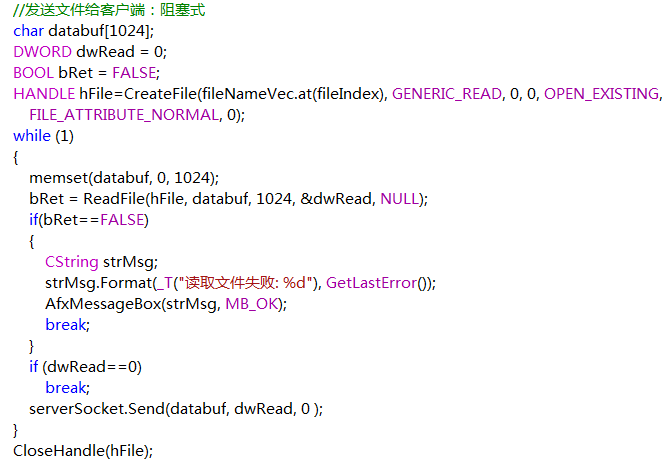
（2）处理客户端socket的线程函数：



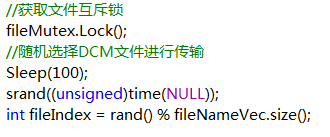
（3）按钮响应函数：



（4）文件发送：



（5）互斥锁的使用：





这里用于随机选取Server目录下的DICOM文件样例。

# 7.附录

## 7.1 使用说明

**（1）传输测试**

1. 首先打开“测试服务端”目录下的“测试服务端.exe”程序，运行模拟服务端；
2. 打开“传输测试.exe”，程序会自动生成“Client”目录作为接受文件的目录；
3. 测试前点击“配置”按钮，输入服务端的IP地址和端口号，单击确定，回到主界面；
4. 点击“传输测试”按钮，进行DICOM文件传输，同时显示接收到的DICOM文件和传输准确率；
5. 点击“退出”按钮退出。

**说明：**

1. 不要删除“测试服务端”目录底下的“Server”目录，它用于存储测试用的DICOM文件样例；
2. 可以删除本目录底下“Client”目录，它用于接收服务端传输过来的DICOM文件；
3. 传输准确率基于DICOM3.0兼容性和传输兼容性两方面来计算。

**（2）压缩测试**

* 1. 首先打开“选择文件”，在弹出文件对话框内选择符合DICOM3.0标准的DICOM文件（若不符合将报错）；
  2. 点击“JPEG无损压缩”按钮进行JPEG无损压缩测试，生成的压缩DICOM文件与源文件在同一目录下；
  3. 点击“JPEG有损压缩”按钮进行JPEG有损压缩测试，生成的压缩DICOM文件与源文件在同一目录下；
  4. 点击“RLE无损压缩”按钮进行RLE无损压缩测试，生成的压缩DICOM文件与源文件在同一目录下；
  5. 压缩同时图片控件显示压缩后的DICOM图像，列表控件显示压缩后的各类参数。

**说明：**

1. “DCM\_samples”目录下有合适的DICOM文件样例。由于Unicode编码问题，采用别的文件目录时注意不能有中文路径；
2. 压缩率可以使用压缩前后的文件大小比值来衡量，也可以使用压缩前后的比特率比值来衡量。

**（3）并发测试**

* 1. 首先打开“测试服务端”目录下的“测试服务端.exe”程序，运行模拟服务端；
  2. 打开“并发测试.exe”，程序会自动生成“Client”目录作为接受文件的目录；
  3. 测试前点击“配置”按钮，输入服务端的IP地址和端口号，单击确定，回到主界面；
  4. 点击“增加客户端”按钮，模拟客户端并发进行DICOM文件传输，同时显示传输效率和传输时间曲线；
  5. 点击“退出”按钮退出。

**说明：**

* 1. 不要删除“测试服务端”目录底下的“Server”目录，它用于存储测试用的DICOM文件样例；
  2. 可以删除本目录底下“Client”目录，它用于接收服务端传输过来的DICOM文件；
  3. 每次增加客户端数目为5个，最大客户端数目为100个；
  4. 传输率以1MB/s为最高，设定显示为100%，实际传输速率按照比例折算成百分比；
  5. 传输时间以10000ms为最高，实际传输时间按照实际显示。

**（4）服务端**

* 1. 点击“启动服务器”按钮来开启模拟服务端，IP地址自动获取本机IP，端口号默认8888，列表控件用于显示消息；
  2. 传输测试和并发测试前均需要开启服务端，压缩测试不需要；
  3. 需要将包含DICOM文件样例的“Server”文件夹放置于同一目录下。

## 7.2 DICOM文件样例

DICOM文件样例来自于：

<http://www.aycan.de/lp/sample-dicom-images.html>

<http://barre.nom.fr/medical/samples/>

## 7.3 DCMTK开源库

DCMTK开源库下载地址：

<http://dicom.offis.de/dcmtk.php.en>