《计算机网络》课程大作业

实验二：编写 SMTP 服务器并观察通信过程

姓名：王科

学号：1310583

专业：计算机科学与技术

完成日期：2015/11/4

一、 实验要求介绍

SMTP 和 POP3 协议是目前电子邮件应用系统中最重要的两个协议。深入理解 SMTP 和 POP3 协议的工作过程对理解整个电子邮件系统具有重要的意义。本实验要求编写一个简化的 SMTP 邮件服务器，通过观察电子邮件应用程序（如 Outlook Express 等）与 SMTP 邮件服务器的交互过程，加深对整个邮件服务系统的理解。

目的：观察电子邮件应用程序与 SMTP 邮件服务器的命令交互过程

要求：简化的 SMTP 服务器

- 响应客户 SMTP 命令，将命令的交互过程和收到的邮件显示到屏幕上
- 支持单用户
- 不保存和转发收到的邮件
- 不作错误处理
- 要求能显示客户端发送的图片。

二、 实验编译运行环境

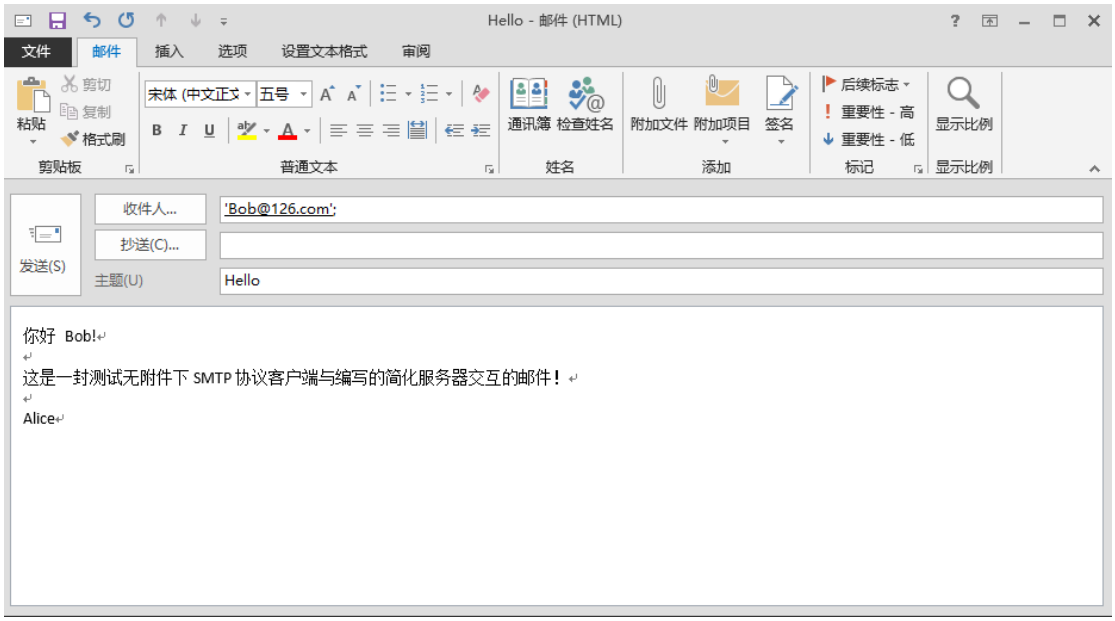
本程序编译环境是：Visual Studio 2012；系统环境是：Windows 8（64 位）；测试用 STMP 邮件客户端是 Outlook Express 2013;

三、 编写简化的 SMTP 服务器运行效果

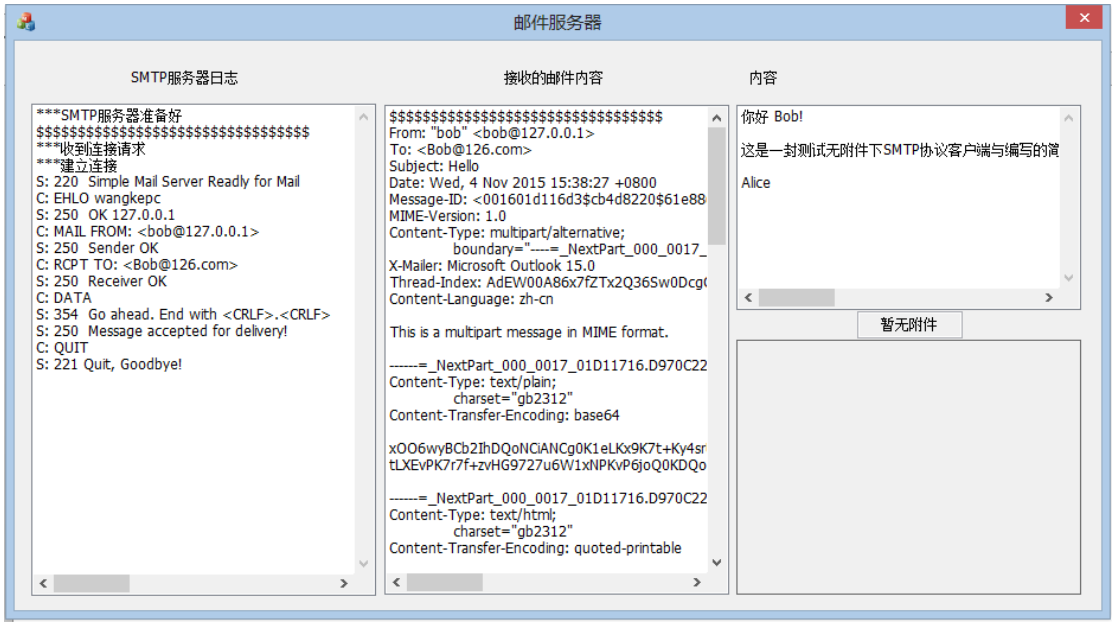
1. 简化 SMTP 服务器界面及功能区介绍

2. **观察发送无附件的邮件**

➢ 在 Outlook Express 邮件客户端发送如下内容邮件：

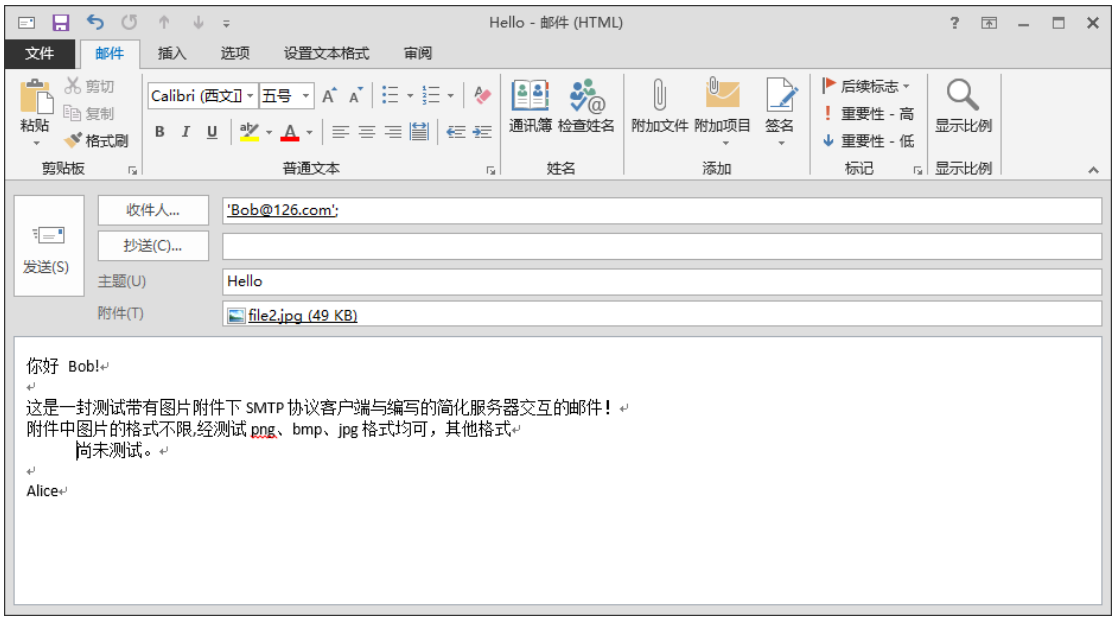


➢ 点击发送过后，在编写的简化 SMTP 邮件客户端上显示如下：



3. **观察发送带有图片附件的邮件**

➢ 在 Outlook Express 邮件客户端发送如下内容邮件



➢ 点击发送过后，在编写的简化 SMTP 邮件客户端上显示如下：

注意：邮件中图片附件在此简化的 SMTP 服务器上只能显示一张；格式为任意格式，经测试常见的图片格式如.png、.jpg、.bmp 均可支持；图片大小可任意大小，但是当图片大小大于 100K 时服务器会见过较长时间才能解码并显示在 picture control 控件中。

四、 实验原理

1. SMTP 服务器日志区：

此部分主要是邮件客户端与服务器建立通讯的过程，在此程序中主要是简单邮件传输协议 SMTP 的实现：

其中，SMTP 服务器在 TCP 的 25 端口守候，其规定发送程序和接收程序之

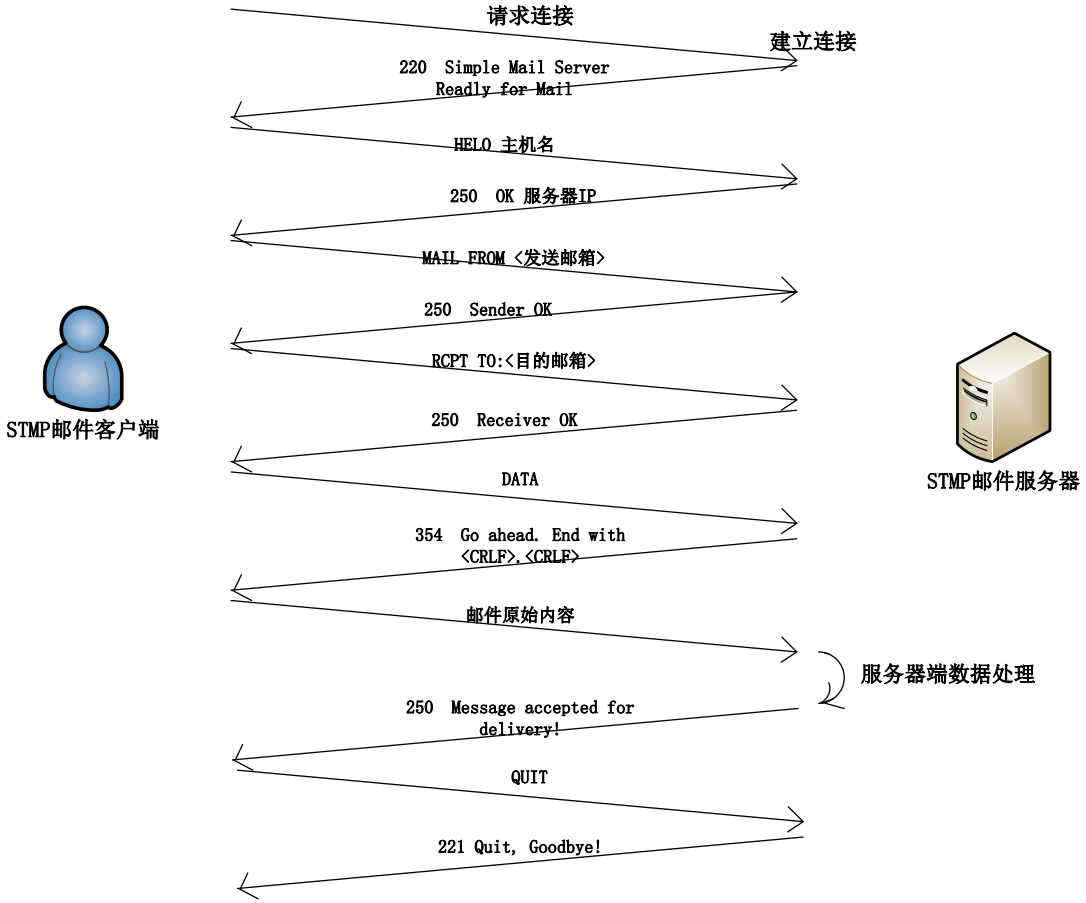
间的命令和应答命令和响应都是可读的 ASCII 字符串。常见的 SMTP 命令有如下：

命令	描述
HELO <主机域名>	开始会话
MAIL FROM：<发送者电子邮件地址>	开始一个邮递处理，指出邮件发送者
RCPT TO：<接收者电子邮件地址>	指出邮件接收者
DATA	接收程序将 DATA 命令后面的数据作为邮件内容处理，直到<CR><LF>.<CR><LF>出现
RSET	中止当前的邮件处理
NOOP	无操作
QUIT	结束会话

其应答响应命令以 3 位数字开始，后面跟有该响应的具体描述，常见的应答命令有如下：

命令	**描述**
220	域服务准备好
221	系统状态或系统帮助应答
250	请求的命令成功完成
354	可以发送邮件内容
500	语法错误，命令不能识别
502	命令未实现
550	邮箱不可用

在本简化 SMTP 邮件服务器中与邮件客户端应答命令和响应命令如下：



2. **邮件数据区：**

此部分进行邮件内容的显示与处理，当接收到邮件内容时，按照’\r\n’分隔开每一行，当遇到’\r\n.\r\n’时结束邮件内容的显示。

3. **邮件正文区：**

此部分将邮件内容中邮件正文显示出来，当在邮件中遇到第一个 ("base64\r\n" 时作为邮件正文 base64 编码的开始处，接着在其后第一次遇到 ("\r\n------") 时作为邮件正文 base64 编码的结束处。将其中间字符串截取出来进行 base64 解码，然后显示到该区域。其中 base64 编码规则如下：

- 将 3 个字节转换成 4 个可打印字符

  将每 3 个字节做为一个整体将其划分为 4 组，每组 6 位

  将 6 位的值作为索引，映射为对应的可打印 ASCII 字符
- 原始文件尾部处理

剩 1 个字节：后面补 4 个比特的“0”，再分成 2 个 6 位组，映射为 2 个 ASCII 字符，而后再填充两个“=”

剩 2 个字节：后面补 2 个比特的“0”，再分成 3 个 6 位组，映射为 3 个 ASCII 字符，而后再填充 1 个“=”

- 添加回车换行：变换后，每 76 个字符后增加一回车换行

4. **附件图片区：**

此部分同样先进行附件图片内容的获取，当在邮件中遇到第一个 `("filename")` 时作为邮件附件图片 base64 编码的开始处，接着在其后第一次遇到 `_T("------=")` 时作为邮件附件图片 base64 编码的结束处。将其中间字符串截取出来进行 base64 解码，写入磁盘文件流，然后使用 CImage 类函数调用显示图片到该区域。

五、 程序实现步骤及代码

1. **SMTP 服务器与客户端应答：**

- OnSend（）发送函数

```
void MySocket::OnSend(int nErrorCode)
{
    // TODO: 在此添加专用代码和/或调用基类
    CMailServerDlg* dlg = (CMailServerDlg*)theApp.m_pMainWnd; //获取对话框句柄

    if(!m_state&&m_count==0){
            //域服务准备好
        char buffff[100]={"220  Simple Mail Server Readly for Mail\r\n"};
        strcpy_s(m_szBuffer,buffff);
        Send(m_szBuffer,strlen(m_szBuffer));
        dlg->mc_server.AddString(_T("S: 220  Simple Mail Server Readly for Mail\r\n"));
        m_state=true;
        AsyncSelect(FD_READ);   //触发OnReceive函数
    }

    if(!m_state&&m_count==1){
            //接收“主机域名”的请求命令成功完成
        memset(m_szBuffer,0,sizeof(m_szBuffer));
        char buffff[100]={"250  OK 127.0.0.1\r\n"};
        strcpy_s(m_szBuffer,buffff);
        Send(m_szBuffer,strlen(m_szBuffer));
        dlg->mc_server.AddString(_T("S: 250  OK 127.0.0.1\r\n"));
        m_state=true;
        AsyncSelect(FD_READ);   //触发OnReceive函数
    }
```

```
    if(!m_state&&m_count==2){
            //接收“发送者电子邮件地址”的请求命令成功完成
        memset(m_szBuffer,0,sizeof(m_szBuffer));
        char buffff[100]={"250  Sender OK\r\n"};
        strcpy_s(m_szBuffer,buffff);
        Send(m_szBuffer,strlen(m_szBuffer));
        dlg->mc_server.AddString(_T("S: 250  Sender OK\r\n"));
        m_state=true;
        AsyncSelect(FD_READ);    //触发OnReceive函数
    }

        if(!m_state&&m_count==3){
            //接收“接收者电子邮件地址”的请求命令成功完成
        memset(m_szBuffer,0,sizeof(m_szBuffer));
        char buffff[100]={"250  Receiver OK\r\n"};
        strcpy_s(m_szBuffer,buffff);
        Send(m_szBuffer,strlen(m_szBuffer));
        dlg->mc_server.AddString(_T("S: 250  Receiver OK\r\n"));
            m_state=true;
        AsyncSelect(FD_READ);    //触发OnReceive函数
    }
```

```
    if(!m_state&&m_count==4){
            //可以发送邮件内容
        memset(m_szBuffer,0,sizeof(m_szBuffer));
        char buffff[100]={"354  Go ahead. End with <CRLF>.<CRLF>\r\n"};
        strcpy_s(m_szBuffer,buffff);
        Send(m_szBuffer,strlen(m_szBuffer));
        dlg->mc_server.AddString(_T("S: 354  Go ahead. End with <CRLF>.<CRLF>\r\n"));
            m_state=true;
        AsyncSelect(FD_READ);    //触发OnReceive函数
    }
```

```
    if(!m_state&&m_count==5){
            //接收“接收者电子邮件地址”的请求命令成功完成
        memset(m_szBuffer,0,sizeof(m_szBuffer));
        char buffff[100]={"250  Message accepted for delivery!\r\n"};
        strcpy_s(m_szBuffer,buffff);
        Send(m_szBuffer,strlen(m_szBuffer));
        dlg->mc_server.AddString(_T("S: 250  Message accepted for delivery!\r\n"));
            m_state=true;
        AsyncSelect(FD_READ);    //触发OnReceive函数
    }

    if(!m_state&&m_count==6){
            //系统状态、系统帮忙应答
        memset(m_szBuffer,0,sizeof(m_szBuffer));
        char buffff[100]={"221 Quit, Goodbye!\r\n"};
        strcpy_s(m_szBuffer,buffff);
        Send(m_szBuffer,strlen(m_szBuffer));
        dlg->mc_server.AddString(_T("S: 221 Quit, Goodbye!\r\n"));
            m_state=true;
        AsyncSelect(FD_READ);    //触发OnReceive函数
    }


    CAsyncSocket::OnSend(nErrorCode);
}
```

- OnReceive（）接收函数

```
void MySocket::OnReceive(int nErrorCode)
{
    // TODO: 在此添加专用代码和/或调用基类
    CMailServerDlg* dlg = (CMailServerDlg*)theApp.m_pMainWnd; //获取对话框
    if(m_state&&m_count!=4){
            //开始会话
        memset(m_szBuffer,0,sizeof(m_szBuffer));
        m_nLength =Receive(m_szBuffer,sizeof(m_szBuffer),0); //接收数据
        CString rec(m_szBuffer);
        dlg->mc_server.AddString(_T("C: ")+rec);//显示
        m_count++;
        m_state=false;
        AsyncSelect(FD_WRITE); //触发OnSend函数
    }
    if(m_state&&m_count==4){
            //在这里接收邮件内容数据
        memset(m_szBuffer,0,sizeof(m_szBuffer));
        m_nLength =Receive(m_szBuffer,sizeof(m_szBuffer),0); //接收数据
        dlg->mc_mail.AddString(_T("$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$"));
```

```
        //转换换行格式显示在ListBox控件中
        //bool con_data=false;      //标记是否进入正文状态
        char content_buff[528192];
        int i_out=0,i_in;
        for(int i=1;;i++){
            memset(content_buff,0,sizeof(content_buff));
            i_in=0;
            while(m_szBuffer[i_out]!='\n'&&m_szBuffer[i_out]!='\0'&&m_szBuffer[i_out]!=0){
                content_buff[i_in++]=m_szBuffer[i_out++];
            }
            i_out++;
            CString rec(content_buff,i_in);
            rec.Remove('\r');
            rec.Remove('\n');
            dlg->mc_mail.AddString(rec);//显示
            m_content_string=m_content_string+rec;
            //以“\r\n.\r\n”结束数据接收
            if((m_szBuffer[i_out]=='.'&&m_szBuffer[i_out+1]=='\r'&&m_szBuffer[i_out+2]=='\n')){
                m_count++;
                m_state=false;
                //邮件正文解码显示
                ShowData();
                //邮件附件显示
                ShowImage();
                break;
            }
            if(m_szBuffer[i_out]==0)
                break;
        };
        AsyncSelect(FD_WRITE); //触发OnSend函数
    }
```

2. 邮件正文区显示：

- 显示邮件正文 ShowData()

```
//显示邮件正文
void MySocket::ShowData()
{
    CMailServerDlg* dlg = (CMailServerDlg*)theApp.m_pMainWnd; //获取对话框句柄
    CString TEXT;       //将要显示在屏幕上的内容
    TEXT.Empty();

    dlg->mc_text.ResetContent();    //清空列表框

    if(m_content_string.Find(_T("Content-Transfer-Encoding: 7bit"))!=-1){
        //正文是7bit编码，无需解码！
        int pos1=m_content_string.Find(_T("Content-Transfer-Encoding: 7bit"))+33;
        int pos2=m_content_string.Find(_T("------"),pos1+1);
        TEXT=m_content_string.Mid(pos1,pos2-pos1-2);
    }
    else{
        if(m_content_string.Find(_T("Content-Transfer-Encoding: base64"))!=-1){
                //正文是base64编码，进行解码！
            CString  mailContent;   //存放正文的base64编码
            m_content_string.Remove('\r');
            m_content_string.Remove('\n');
            int pos1=m_content_string.Find(_T("Content-Transfer-Encoding: base64"))+33;
            int pos2=m_content_string.Find(_T("------"),pos1+1);
            mailContent=m_content_string.Mid(pos1,pos2-pos1);
            if(mailContent.IsEmpty())
                dlg->mc_text.AddString(L"邮件正文解析错误！");
            //删除'\r\n';
            int mailContent_size;
            TEXT=Text_DecodeBase64(mailContent,&mailContent_size);
        }
    }
```

```
    if(!TEXT.IsEmpty()){
    //显示在屏幕上
        int pos_begin=0,pos_end=TEXT.Find('\r',1);
        while(pos_end!=-1)
        {
            CString temp=TEXT.Mid(pos_begin ,pos_end - pos_begin );
            dlg->mc_text.AddString(temp);
            pos_begin=pos_end+4;
            pos_end=TEXT.Find('\r',pos_begin+1);
        }
    }
}
```

3. **邮件附件图片显示：**

- 显示图片 ShowImage()

```
//显示图片
void MySocket::ShowImage(void)
{
    CMailServerDlg* dlg = (CMailServerDlg*)theApp.m_pMainWnd; //获取对话框句柄
    CString  ImageContent;  //存放正文base64编码
    int pos1=m_content_string.Find(_T("filename"));
    if(pos1==-1)
        return;
    int pos2=m_content_string.Find('"',pos1+7);
    int pos3=m_content_string.Find('"',pos2+1);
    m_filename=m_content_string.Mid(pos2+1,pos3-pos2-1);    //获取附件名称
    dlg->m_filename=m_filename;
    dlg->UpdateData(false);                                 //显示附件名称在控件中
    int pos4=m_content_string.Find(_T("------="),pos3+1);
    ImageContent=m_content_string.Mid(pos3+1,pos4-pos3-1);

    std::vector<char> bytes;
    Image_DecodeBase64(ImageContent, bytes);
    std::fstream of(m_filename, std::ios_base::out | std::ios_base::binary);;
    of.write(static_cast<const char*>(&bytes[0]), bytes.size());
    of.close();


    //显示附件图片
    if(!m_image.IsNull()) {  //确定位图当前是否正在加载
        m_image.Detach();
    }
```

```
    m_image.Load(m_filename);
    if (!m_image.IsNull()) {

        SetStretchBltMode(dlg->mc_picture.GetDC()->GetSafeHdc(), HALFTONE);
        CRect dest;
        dlg->mc_picture.GetClientRect(&dest);

        float nRatioImage = m_image.GetWidth() / static_cast<float>(m_image.GetHeight());
        float nRatioDest = dest.Width() / static_cast<float>(dest.Height());
        CRect rectDraw = dest;
        if (nRatioImage > nRatioDest) {    //设置在空间居中显示
            rectDraw.SetRect((dest.Width()/2)-(rectDraw.right/2),(dest.Height()/2)-
                ((rectDraw.right / nRatioImage)/2),(dest.Width()/2)+(rectDraw.right/2)
                ,(dest.Height()/2)+((rectDraw.right / nRatioImage)/2) );
        }
        else if (nRatioImage < nRatioDest) {
            rectDraw.SetRect((dest.Width()/2)-((rectDraw.bottom * nRatioImage)/2),
                (dest.Height()/2)-( rectDraw.bottom/2), (dest.Width()/2)+
                ((rectDraw.bottom * nRatioImage)/2),(dest.Height()/2)+( rectDraw.bottom/2));
        }
        m_image.Draw(dlg->mc_picture.GetDC()->GetSafeHdc(), rectDraw);
    }


}
```

六、 实验总结

通过本次实验学习了如何使用 CAsyncSocket 类编写一个简单 SMTP 邮件服务器程序，加深了 TCP 建立连接和 SMTP 协议的理解。通过观察电子邮件应用程序（如 Outlook Express 等）与 SMTP 邮件服务器的交互过程，加

深对整个邮件服务系统的理解。

总之，这次编程使我对于网络协议的设计和通信有了更深的认识，提高了对这门课程的兴趣，增强了动手编程能力。