实 验 十一　基于 SMTP 协议的邮件发送

11.1　实验目的

电子邮件是互联网中重要的一种应用，给人们提供了极大便利，本实验主要介绍基于 SMTP 协议实现邮件的发送，邮件数据须经过 BASE64 编码，本实验采用了.NET 平台提供的 MailMessage 类来简化编码。

11.2　SMTP 协议

简单邮件传输协议 (Simple Mail Transfer Protocol, SMTP) 是互联网普遍应用的 email 传输标准，它属于 TCP/IP 协议族使用 TCP 端口 25。SMTP 是简单的基于文本的协议，在其之上指定了一条消息的一个或多个接收者（在大多数情况下被确认是存在的），然后消息文本会被传输。SMTP 要为一个给定的域名决定一个 SMTP 服务器，需要使用 MX (Mail eXchange) DNS。

SMTP 协议最初实现基于纯 ASCII 文本传输，二进制文件不被支持，MIME 标准设计用来编码二进制文件，MIME 里使用 BASE64 编码，它要求把每三个 8Bit 的字节转换为四个 6Bit 的字节（3Œ8 = 4Œ6 =24），然后把 6Bit 再添两位高位 0，组成四个 8Bit 的字节，这样编码后的数据比原始数据略长，为原来的 4/3。完整的 base64 定义可见 RFC1421 和 RFC2045。编码后的数据变成文本文件即可通过 SMTP 来传输。最初的 SMTP 的局限之一在于它没有对发送方进行身份验证的机制，过多垃圾邮件会给服务器造成负担，后来定义了 SMTP-AUTH 扩展，它要求用户必须进行身份登录后才可以执 SMTP 命令。

可以通过 telnet 程序测试一个 SMTP 服务器，方法是在命令行输下面的命令：

telnet smtp.126.com 25

成功连接服务器后，输入 SMTP 的命令进行邮件操作。

11.3　实验内容

11.3.1　SMTP 协议发送邮件

SMTP 的命令和响应都是基于文本，以命令行为单位，换行符为 CR/LF。响应信息一般只有一行，由一个 3 位数的代码开始，后面可附上简短的文字说明。SMTP 要经过建立连接、传送邮件和释放连接 3 个阶段。TCP 连接 SMTP 服务器发送邮件的基本流程如下：

1. 建立 TCP 连接。
2. 客户端向服务器发送 HELO 命令
3. 发送 USER 和 PASS 命令以进行身份验证。

表 11-1 SMTP 基本命令

命令	参数	描述
USER	用户名	与 pass 命令一起确认用户信息
HELO	无	问候消息，询问服务器支持何种 SMTP 扩展。
AUTH LOGIN	无	用户认证命令，准备输入用户名和密码
USER	用户名	base64 算法加密后的用户名字符串
PASS	密码	base64 算法加密后的密码字符串
MAILFROM	发价人地址	指定发件人地址。
RCPTTO	收件人地址	指定收件人地址，可有多个。
DATA	无	在单个或多个 RCPT 命令后，表示所有的邮件接收人已标识，并初始化数据传输，DATA 命令后的内容将是按照 MIME 协议定义的邮件数据内容，邮件数据以"." 结束。
VRFY	无	用于验证指定的用户/邮箱是否存在；由于安全方面的原因，服务器常禁止此命令
HELP	无	查询服务器支持什么命令
NOOP	无	无操作，服务器应响应 OK
QUIT	无	结束会话，退出登录，服务器断开连接

4. 发送 MAILFROM 命令标识发送者，RCPTTO 命令标识接收者。
5. 发送命令 DATA 准备发送邮件内容。
6. 输入邮件内容并以"." 结束。
7. 用 QUIT 命令退出结束会话。

SMTP 身份验证的用户名与密码以及邮件正文都需要进 BASE64 进行编码，不方便直接输入邮件信息。C# 中可使用.NET 平台 Convert 类的 ToBase64String 方法实现将字节数据转换为 BASE64 字符串。.NET 平台的 SmtpClient、MailAddress 和 MailMessage 类进一步方便进行用户身份验证和邮件数据的生成，用户不但无需创建 Socket 连接也免去了编码的运算。使用这三个类的步骤如下：

1. 创建 SmtpClient 对象，指定服务主机。
2. 通过 SmtpClient 对象设置账户登录信息。
3. 使用类 MailAddress 设置邮件发送者和接收者地址。
4. 创建 MailMessage 对象，设置邮件内容和文本编码方式，邮件标题，附件。
5. 定义 SmtpClient 对象邮件发送完成的回调函数。
6. 调用 SmtpClient 对象的邮件发送方法。

采 SmtpClient、MailAddress 和 MailMessage 三个类进行邮件发送的示例线程代码如下，这个示例线程简化了用户名和密码以及邮件正文的 BASE64 编码过程。

```
static void thr_smtp_con()
{
  SmtpClient client = new SmtpClient("smtp.126.com");
  client.UseDefaultCredentials = true;
  client.Credentials = new System.Net.NetworkCredential("zscleonet","goodstudent");
```

```
    client.DeliveryMethod = SmtpDeliveryMethod.Network;
    MailAddress from = new MailAddress("zscleonet@126.com",
    " 李老师" + (char)0xD8 + " 中山学院",
    System.Text.Encoding.UTF8);
    MailAddress to = new MailAddress("zscleo@126.com");
    MailMessage mailMessage = new MailMessage(from, to);
    mailMessage.Body = "This is a test e-mail message sent by an application. ";
    // Include some non-ASCII characters in body and subject.
    string someArrows = new string(new char[] { '\u2190', '\u2191', '\u2192', '\u2193' });
    mailMessage.Body += Environment.NewLine + someArrows;
    mailMessage.BodyEncoding = System.Text.Encoding.UTF8;
    mailMessage.Subject = "test message 1" + someArrows;
    mailMessage.SubjectEncoding = System.Text.Encoding.UTF8;
    client.SendCompleted += new
    SendCompletedEventHandler(SendCompletedCallback);
    string userState = "test message1";
    send_str = " 开始发送邮件 \r\n";
    // current working directory.
    string attachfile = "Kid1.png";
    Attachment atcData = new Attachment(attachfile, MediaTypeNames.Application.Octet);
    // Add time stamp information for the file.
    ContentDisposition disposition = atcData.ContentDisposition;
    disposition.CreationDate = System.IO.File.GetCreationTime(attachfile);
    disposition.ModificationDate = System.IO.File.GetLastWriteTime(attachfile);
    disposition.ReadDate = System.IO.File.GetLastAccessTime(attachfile);
    // Add the file attachment to this e-mail message.
    mailMessage.Attachments.Add(atcData);
    SendMessage(main_wnd_handle, UPDATE_INFO, 100, 100);
    client.SendAsync(mailMessage, userState);
    MRE_check_end.WaitOne();
    send_str = " 邮件发送完成 \r\n";
    SendMessage(main_wnd_handle, UPDATE_INFO, 100, 100);
    message.Dispose();
}
```

邮件发送响应的回调函数如下：

```
private static void SendCompletedCallback(object sender, AsyncCompletedEventArgs e)
{
    String token = (string)e.UserState;
    if (e.Cancelled)
    {
```

```
        send_str = " 邮件发送取消 \r\n";
        SendMessage(main_wnd_handle, UPDATE_INFO, 100, 100);
      }
      if (e.Error != null)
      {
        send_str = string.Format("[{0}] {1}", token, e.Error.ToString());
        SendMessage(main_wnd_handle, UPDATE_INFO, 100, 100);
      }
      else
      {
        send_str = " 邮件已发送 \r\n";
        SendMessage(main_wnd_handle, UPDATE_INFO, 100, 100);
      }
      MRE_check_end.Set();
    }
```

11.4 实验作业

1. 采用本实验内容，编写一个邮件收发的程序。
2. 根据 SMTP 协议，不使用 SmtpClient 类重新设计程序实现邮件发送。