

MINISTERUL EDUCATIEI REPUBLICII MOLDOVA
UNIVERSITATEA TEHNICA A MOLDOVEI
Facultatea „Calculatoare, Informatică și Microelectronică”
Departamentul Automatica și Tehnologii Informaționale

RAPORT

Programarea în rețea

Lucrare de laborator Nr. 4

Tema: Protocoalele poștei electronice

A elaborat:

st. gr. TI-142 Țurcan Tudor

A verificat:

asist. univ. Donos Eugenia

Chișinău 2017

Cuprins

Scopul lucrării	3
Sarcina lucrării.....	3
Considerații teoretice.....	3
Desfășurarea lucrării	4
Concluzie.....	10
Bibliography	11

Scopul lucrării

Protocolul SMTP de transfer mesaje electronice. Comenzile protocolului SMTP. Comenzile POP3 de autorizare, tranzacție și actualizare. Componente .Net/MFC pentru elaborare de poștă electronică.

<https://github.com/TurcanTudor/Programarea-in-Retea.git>

Sarcina lucrării

De creat o aplicație care permite transmiterea mesajelor de email către alți utilizatori, și de accesat email-ul personal pentru descărcarea listei de email-uri și ștergerea unui mesaj de la alt utilizator.

Considerații teoretice

Simple Mail Transfer Protocol (pe scurt, SMTP; în traducere aproximativă Protocolul simplu de transfer al corespondenței) este un protocol simplu, folosit pentru transmiterea mesajelor în format electronic pe [Internet](#). SMTP folosește portul de aplicație 25 [TCP](#) și determină adresa unui [server](#) SMTP pe baza înregistrării [MX](#) din configurația [serverului DNS](#).

Protocolul SMTP specifică modul în care mesajele de poștă electronică sunt transferate între [procese](#) SMTP aflate pe sisteme diferite. Procesul SMTP care are de transmis un mesaj este numit client SMTP iar procesul SMTP care primește mesajul este serverul SMTP. Protocolul nu se referă la modul în care mesajul ce trebuie transmis este trecut de la utilizator către clientul SMTP, sau cum mesajul recepționat de serverul SMTP este livrat utilizatorului destinatar și nici cum este memorat mesajul sau de câte ori clientul SMTP încearcă să transmită mesajul.

Un server SMTP trebuie să cunoască cel puțin următoarele comenzi :

- HELO - identificare computer expeditor;
- EHLO - identificare computer expeditor cu cerere de mod extins;
- MAIL FROM - specificarea expeditorului;
- RCPT TO - specificarea destinatarului ;
- DATA - conținutul mesajului;
- RSET – Reset;
- QUIT - termină sesiunea;

- HELP - ajutor pentru comenzi;
- VRFY - verifica o adresa;
- EXPN - expandează o adresa;
- VERB - informații detaliate.

IMAP (Internet Message Access Protocol) a fost dezvoltat inițial în 1986 ca o alternativă la **POP**. Diferența majoră dintre IMAP și POP este aceea că mesajele sunt prelucrate direct pe server. Clientul de email aduce pe calculatorul sau terminalul inteligent ***o copie*** a mesajului, însă toate operațiunile (modificare, ștergere, organizare în folder) sunt efectuate direct pe serverul IMAP. De aici și avantajul major: folosind protocolul IMAP, poți accesa mesajele de pe mai multe terminale (calculatorul de la birou, laptopul de acasă, tableta portabilă sau telefonul inteligent), fără grija pierderii unor mesaje importante. Păstrând mesajele pe server, dezavantajul tău este acela al spațiului scump (spațiul de stocare locat este mult mai ieftin decât spațiul pe serverul furnizorului). Acest dezavantaj este însă estompat de mai toate programele de email, care permit arhivarea sau backup-ul local al mesajelor.

Desfășurarea lucrării

Unul din serviciile utilizate în zilele noastre este poșta electronică, care deși pare demodată se utilizează pe larg în diferite companii, de persoane fizice și de un număr mare de utilizatori.

Poșta electronică utilizează 2 protocoale de bază, acestea fiind SMTP și POP3, ulterior înlocuit de protocolul IMAP. Anume aceste protocoale menționate mai sus vor fi implementate în aplicația noastră. Cu ajutorul protocolului SMTP se vor transmite mesaje, în continuare denumite email-uri către un receptor, iar cu ajutorul protocolului IMAP vom extrage mesajele de pe poșta electronică și le vom afișa în aplicația noastră.

Pentru aplicație s-a utilizat ca limbaj de programare, C#. Folosind clasa predefinită *SmtplibClient* putem transmite email-uri către alți utilizatori de poșta electronică. Creăm o variabilă de tipul clasei menționate mai sus, care o instanțiem cu ajutorul constructorului predefinit, unde indicăm ca parametru serverul SMTP care dorim să-l accesăm. Astfel are loc stabilirea unei conexiuni dintre emițătorul SMTP și receptorul SMTP (1).

Variabila noastră posedă attributele clasei noastre instanțiate, cum ar fi adresa emițătorului, adresă receptorului, subiectul mesajului transmis și desigur corpul mesajului nostru. Toate acestea pot fi observate în figura 1 de mai jos.

```
var smtpServer = new SmtpClient("smtp.gmail.com");  
var mail = new MailMessage();  
  
mail.From = new MailAddress(txtemail.Text);  
mail.To.Add(txtreceiver.Text);  
mail.Subject = txtsubject.Text;  
mail.Body = txtbody.Text;  
smtpServer.Port = 587;  
smtpServer.UseDefaultCredentials = false;  
smtpServer.EnableSsl = true;
```

Fig. 1 Metodele Clasei *SmtpClient*

Un atribut ce trebuie menționat, este *Port* unde se indică portul prin care vor avea loc tranzacțiile SMTP. Numărul portului ne indică socketul unde trebuie să fie transmis procesul nostru. (2) Astfel cu ajutorul porturilor diferite procese pe același sistem de calcul permit partajarea rețelei simultan.

Totodată la emailul nostru se poate de atașat și un fișier de orice tip, compatibil desigur cu cerințele protocolului nostru (figura 2).

```
System.Net.Mail.Attachment attachment;  
attachment = new System.Net.Mail.Attachment(txtattachment.Text);  
mail.Attachments.Add(attachment);
```

Fig. 2 Atașarea unui fișier

După ce are loc verificarea dacă a fost atașat un fișier sau nu la mesaj, atribuim acreditări emițătorului pentru autentificare. Aici se include adresa de email și parola acestuia.

Metoda *Sent()* trimite mesajul nostru către serverul nostru SMTP pentru livrarea ulterioară la receptor. Dacă mesajul a fost transmis cu succes se afișează un mesaj de confirmare (figura 3).

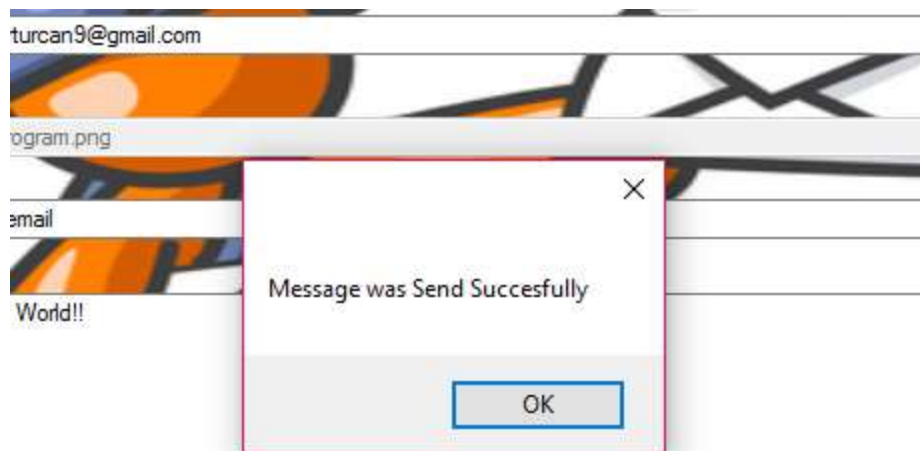


Fig. 3 Trimiterea unui email

În figura de mai jos (figura 4) se vede că mesajul ajuns la destinatar cu succes.

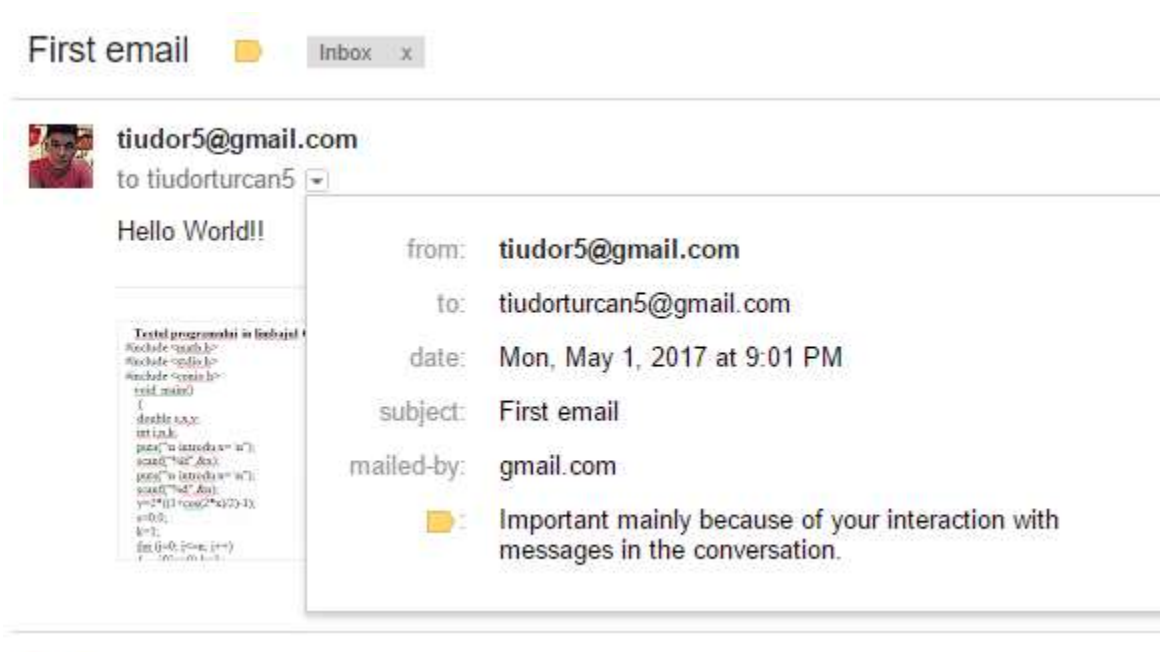


Fig. 4 Verificarea poștei receptorului

Pentru accesarea unui email se utilizează protocolul IMAP (Internet Message Access Protocol). Acesta permite unui client să acceseze și să manipuleze emailurile electronice pe un server. Astfel manipularea mesajelor este echivalentă cu funcționalitatea mapelor locale. (3) Unele funcționalități a acestui protocol sunt crearea, editarea, ștergerea și redenumirea mesajelor, setarea și ștergerea steagurilor.

Pentru accesarea poștei electronice am folosit biblioteca IMapX. (4) Acesta este o bibliotecă open-source. Am creat o metoda *LoginMethod()* care are o variabila de tip *Client* asupra căruia aplicăm metoda *Connect()*. (5) Dacă s-a stabilit conexiunea se va afișa un mesaj de confirmare (figura 5).

```
public void LoginMethod()
{
    var result = false;
    result = Client.Connect();
    if (result)
    {
        MessageBox.Show("Connection Established");
    }
    result = Client.Login(txtEmailId.Text, txtPasswrld.Text);
    MessageBox.Show(result ? "Logged in!" : "Login Failed!!!");
}
```

Fig. 5 Metoda Login pentru stabilirea legăturii dintre Aplicație și server

Pentru vizualizarea mesajelor din poșta electronică avem nevoie de preluat emailurile și de afișat într-un text box. Totodată afișăm subiectul și corpul mesajelor noastre (figura 6).

```

FolderCollection folders = Client.Folders;
foreach (Folder myfolder in folders)
{
    var message = Client.Folders.Sent.Search("ALL");
    foreach (var message in message)
    {
        if (message.Subject.Contains(txtsubject.Text))
        {
            txtListEmail.Text = message.Subject;
            txtContent.Text = message.Body.Text;
        }
    }
}

```

Fig. 6 Metoda de găsire a mesajelor

The screenshot shows a web application interface for email management. At the top, there are two tabs: "SendMail" and "ReceiveMail". Below the tabs, there are input fields for "Email ID:" (containing "tiudor5@gmail.com") and "Password:" (containing "*****"). To the right of these fields are three buttons: "Login", "RetrieveEmail" (which is highlighted with a blue dashed border), and "Delete Email". Below the login section, there are two main panels. The left panel, titled "Inbox Email", contains a list with one item: "First email". The right panel, titled "Message Content", displays the text "Hello World!!".

Fig. 7 Afișarea mesajelor în aplicație

Ștergerea mesajelor are loc prin căutarea în mapa Inbox a mesajelor răspunse deja. Mai apoi cu metoda *Remove()* se șterg mesajele noastre (figura 8).


```
private void btnDelete_Click(object sender, EventArgs e)
{
    var message = Client.Folders.Sent.Search("ALL");
    foreach (var mesage in message)
    {
        if (mesage.Subject.Contains(txtsubject.Text))
        {
            mesage.Remove();
        }
    }

    MessageBox.Show("Message deleted!!!");
}
```

Fig. 8 Funcția de ștergere

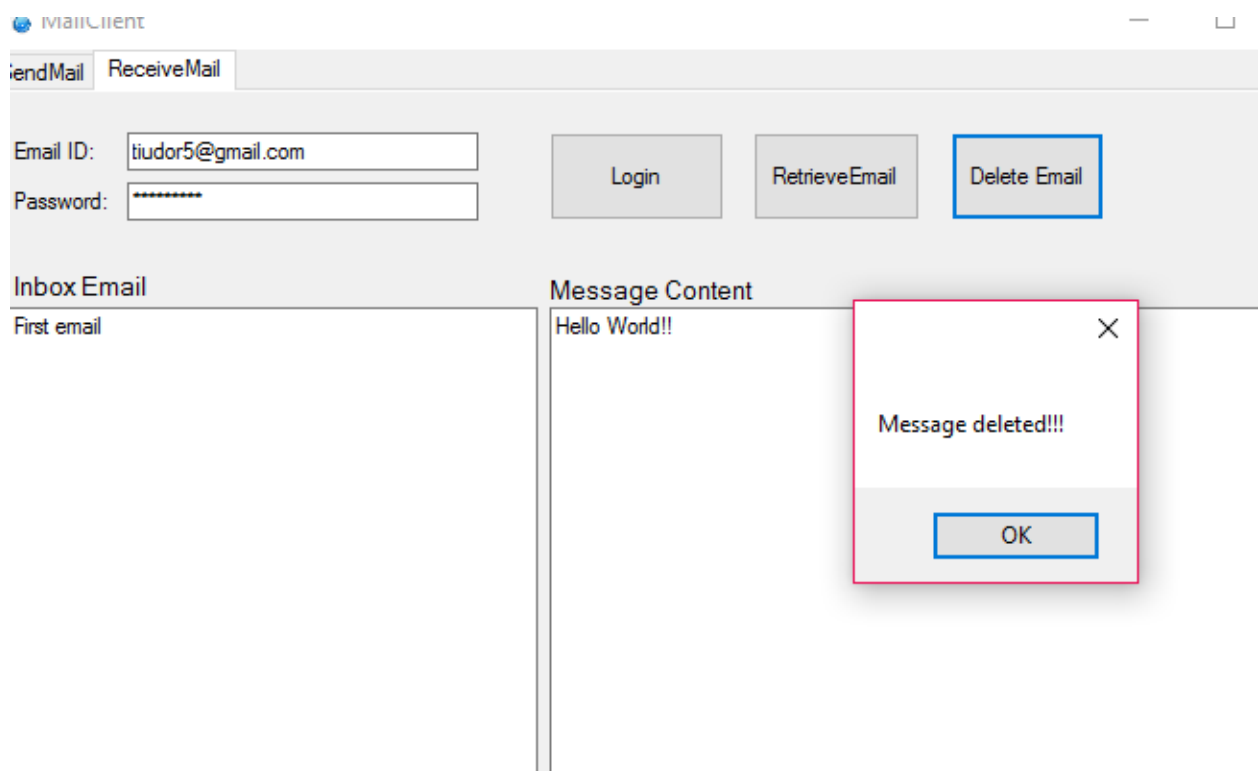


Fig. 9 Ștergerea mesajelor

Concluzie

Deși în ultimii anii poșta electronică scade din popularitate, aceasta este un serviciu care este utilizat pe larg încă de un număr mare de utilizatori. Luând în considerație faptul dat este important de studiat protocoalele pe care rulează acest serviciu.

Astfel cum am menționat în raport SMTP de utilizează pentru transmiterea mesajelor, iar POP3 și IMAP se folosesc pentru manipularea mesajelor pe mașină de calcul. SMTP folosește portul standard TCP= 25 dar portul folosit de gmail este 587. IMAP folosește portul cu numărul 143, iar pentru o conexiune mai securizată utilizează portul 993. În principiu cu ajutorul porturilor se ajunge la posibilitatea de partaja rețeaua pe o mașină de calcul.

Studiind aceste 2 protocoale, și anume particularitățile lor am putut crea o aplicație care ne permite transmiterea de mesaje și totodată manipularea acestora pe o mașină locală.

Bibliography

1. **J.Klensin.** Simple Mail Transfer Protocol. *www.ietf.org*. [Online] [Cited: Aprilie 1, 2016.] <https://www.ietf.org/rfc/rfc2821.txt>.
2. **Quora, Inc.** What is port number and port address. *www.quora.com*. [Online] [Cited: Aprilie 1, 2016.] <https://www.quora.com/What-is-port-number-and-port-address>.
3. **Network Working Group, M.Crispin.** Internet Message Access Protocol - Version 4rev1. *tools.ietf.org*. [Online] March 2003. [Cited: Aprilie 2, 2016.] <https://tools.ietf.org/html/rfc3501#section-6.4.5>.
4. **2006-2016 Microsoft.** ImapX Documentation. *imapx.codeplex.com*. [Online] 2006-2016. [Cited: Aprilie 3, 2016.] <https://imapx.codeplex.com/documentation>.
5. —. Using a login and password to authenticate. *imapx.codeplex.com*. [Online] 2006-2016. [Cited: Aprilie 2, 2016.] <https://imapx.codeplex.com/wikipage?title=Using%20a%20login%20and%20password&referrerTitle=Authentication>.