Ministerul Educaţiei al Republicii Moldova Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatica si Microelectronica Departamentul Ingineria Software si Automatica

**RAPORT**

Lucrarea de laborator nr.4 *Programarea in retea*

## *Tema:* Elemente ale procesarii concurente

A efectuat: st.gr. SI-171 , Buga Grigore

A verificat:

asist. univ. Cebotari Daria

Chisinau 2020

Scopul lucrarii:

Realizarea firelor de execuţie în Java/C#. Proprietăţile firelor. Stările unui fir de execuţie. Lansarea, suspendarea şi oprirea unui fir de execuţie. Grupuri de Thread-uri. Elemente pentru realizarea comunicării şi sincronizării.

Sarcina:

Mail Protocols

Am efectuat un program care implementeaza schimbul de informatii prin protocoalele petru email SMTP si POP3

Am folosit Gmail ca server de mail

Functionalitati implementate

SMTP - trimite un mesaj de pe un email pe altul

POP3 - returneaza datele despre ultimele 10 scrisori de pe mail (Subiect, Expeditor, Mesaj)

Cod sursa.

Mailprotocol.java

import javax.mail.\*;

import javax.mail.internet.InternetAddress;

import javax.mail.internet.MimeBodyPart;

import javax.mail.internet.MimeMessage;

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.Properties;

import java.awt.event.ActionEvent;

import java.util.Properties;

import javax.mail.Folder;

import javax.mail.Message;

import javax.mail.MessagingException;

import javax.mail.NoSuchProviderException;

import javax.mail.Session;

import javax.mail.Store;

public class Mailprotocol {

public static void main(String[] args) {

JFrame frame = new JFrame("Laborator nr.4");

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(300, 200);

JPanel panel = new JPanel();

frame.add(panel);

JButton button = new JButton("SMTP");

JButton button1 = new JButton("POP3");

panel.add(button);

panel.add(button1);

button.addActionListener(new Smtp());

button1.addActionListener(new POP3());

}

static class Smtp implements ActionListener {

@Override

public void actionPerformed(ActionEvent e) {

final String username = "kenny.mccormick443@gmail.com ";

final String password = "kenny443";

Properties props = new Properties();

props.put("mail.smtp.auth", "true");

props.put("mail.smtp.starttls.enable", "true");

props.put("mail.smtp.host", "smtp.gmail.com");

props.put("mail.smtp.port", "25");

Session session = Session.getInstance(props,

new javax.mail.Authenticator() {

protected PasswordAuthentication getPasswordAuthentication() {

return new PasswordAuthentication(username, password);

}

});

try {

Message message = new MimeMessage(session);

message.setFrom(new InternetAddress("kenny.mccormick443@gmail.com"));

message.setRecipients(Message.RecipientType.TO,

InternetAddress.parse("mihai.druta98@gmail.com"));

message.setSubject("Subiectul mesajului ");

message.setText("Mesaj SMTP trimis cu succes");

Transport.send(message);

System.out.println("Executat cu succes");

} catch (MessagingException z) {

throw new RuntimeException(z);

}

}

}

static class POP3 implements ActionListener {

@Override

public void actionPerformed(ActionEvent e) {

String host = "pop.gmail.com";

String mailStoreType = "pop3";

String username = "kenny.mccormick443@gmail.com";

String password = "kenny443";

verid(host, mailStoreType, username, password);

}

private void verid(String host, String mailStoreType, String username, String password) {

try {

//create properties field

Properties properties = new Properties();

properties.put("mail.pop3.host", host);

properties.put("mail.pop3.port", "110");

properties.put("mail.pop3.starttls.enable", "true");

Session emailSession = Session.getDefaultInstance(properties);

//create the POP3 store object and connect with the pop server

Store store = emailSession.getStore("pop3s");

store.connect(host, username, password);

//create the folder object and open it

Folder emailFolder = store.getFolder("INBOX");

emailFolder.open(Folder.READ\_ONLY);

// retrieve the messages from the folder in an array and print it

Message[] messages = emailFolder.getMessages();

System.out.println("Total messages is " + messages.length);

for (int i = 0, n = messages.length; i < n; i++) {

if (i == 10) {break;}

Message message = messages[i];

System.out.println("\n");

System.out.println("Email Number " + (i + 1));

System.out.println("Subject: " + message.getSubject());

System.out.println("From: " + message.getFrom()[0]);

System.out.println("Text: " + message.getContent());

}

//close the store and folder objects

emailFolder.close(false);

store.close();

} catch (NoSuchProviderException e) {

e.printStackTrace();

} catch (MessagingException e) {

e.printStackTrace();

} catch (Exception e) {

e.printStackTrace();

}

}

}

}

Raspunsuri:

• **Cum se poate verifica dacă serverul SMTP funcționează utilizind linia de comandă?**

telnet mailhost.domeniu.ro 25

• **Care sunt comenzile SMTP**

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 - informatii detaliate.

• **Pentru ce este nevoie de MUA, MSA, MTA și MDA**

MUA (agent utilizator de e-mail)

Aplicație client care permite primirea și trimiterea de e-mailuri. Poate fi o aplicație desktop cum ar fi Microsoft Outlook / Thunderbird / ... sau bazată pe web, cum ar fi Gmail / Hotmail / ... (aceasta din urmă se numește și Webmail).

MSA (Agent de trimitere prin poștă)

Un program de server care primește mail de la un MUA, verifică eventualele erori și îl transferă (cu SMTP) către MTA găzduit pe același server.

MTA (agent de transfer de poștă)

O aplicație server care primește mail de la MSA sau de la un alt MTA. Acesta va găsi (prin servere de nume și DNS) înregistrarea MX din zona DNS a domeniului destinatar pentru a ști cum să transferați poșta. Se transferă apoi poșta (cu SMTP) către un alt MTA (cunoscut sub numele de retransmisie SMTP) sau, dacă a fost atins serverul destinatarului, către MDA.

MDA (agent de livrare prin poștă)

Un program de server care primește mail de la MTA-ul serverului și îl stochează în căsuța poștală. MDA este, de asemenea, cunoscut sub numele de LDA (Local Delivery Agent).

Un exemplu este Dovecot, care este în principal un server POP3 și IMAP, care permite unui MUA să recupereze poștă, dar include și un MDA care preia poștă de la un MTA și îl livrează la căsuța poștală a serverului.

• **Care este diferența dintre porturile 25, 465 și 587 ?**

Portul TCP 25 se folosește pentru comunicate SMTP, dar porturile 465 și 587 sunt utilizate ca porturi de transmitere pentru Mail Client

**• Care este diferența dintre porturile 110 și 995 ?**

Pe porturile 110 și 995 lucrează protocolul POP3

Port 110 - acesta este portul necriptat POP3 implicit

Port 995 - acesta este portul pe care trebuie să-l utilizați dacă doriți să vă conectați folosind POP3 în siguranță

**• Care este diferența dintre porturile 143 și 993 ?**

Pe porturile 110 și 995 lucrează protocolul IMAP

Port 110 - acesta este portul necriptat IMAP implicit

Port 995 - acesta este portul pe care trebuie să-l utilizați dacă doriți să vă conectați folosind IMAP în siguranță

**• Cum funcționează protocolul SMTP ?**

Comunicarea intre client și server se realizeaza prin texte ASCII. Inițial clientul stabilește conexiunea către server și așteaptă ca serverul să-i răspundă cu mesajul “220 Service Ready” . Dacă serverul e supraîncărcat, poate să întarzie cu trimirea acestui raspuns. Dupa primirea mesajului cu codul 220 , clientul trimite comanda HELO prin care isi va indica identitatea. In unele sisteme mai vechi se trimite comanda EHLO, comanda EHLO indicand faptul că expeditorul mesajului poate sa proceseze extensiile serviciului și dorește să primească o listă cu extensiile pe care le suportă serverul. Dacă clientul trimite EHLO iar serverul îi răspunde ca aceasta comandă nu e recunoscută, clientul va avea posibilitatea să revină și să trimită HELO.

**• Scopul protocoalelor POP3 și IMAP**

Ambele sunt modalități de conectare la serverul de poștă, astfel încât să puteți citi e-mailurile dvs. printr-un client de e-mail.

**• Diferența dintre POP3 și IMAP**

Cu IMAP, mesajul nu rămâne pe dispozitivul local, cum ar fi un computer, ci rămâne pe server.

Cu e-mail POP3, acesta se va conecta și va încerca să mențină poștă localizată pe dispozitivul local (computer sau mobil).

**• Cum să verificați dacă există o adresă de e-mail fără a trimite un e-mail ?**

Să încercăm să verificăm dacă adresa someuser@gmail.com există sau nu. În primul rând, trebuie să găsim înregistrările MX asociate domeniului destinatarului, în cazul nostru, gmail.com. Vom folosi o utilitate de căutare DNS numită dig care este instalată pe majoritatea sistemelor Linux. La promptul de comandă, tastați:

$ dig gmail.com MX

Rezultatul ar trebui să arate cam așa:

; << >> DiG 9.8.2rc1-RedHat-9.8.2-0.47.rc1.en6 << >> gmail.com MX @8.8.8.8 ;; opțiuni globale: + cmd ;; Am raspuns:; ->> HEADER <<- opcode: QUERY, starea: NOERROR, id: 32294 ;; steaguri: qr rd ra; QUERY: 1, RĂSPUNS: 5, AUTORITATEA: 0, ADDITIONAL: 0 ;; ÎNTREBARE SECȚIUNE:; gmail.com. IN MX ;; RĂSPUNS SECȚIUNEA: gmail.com. 3599 în MX 10 alt1.gmail-smtp-in.l.google.com. gmail.com. 3599 în MX 20 alt2.gmail-smtp-in.l.google.com. gmail.com. 3599 în MX 30 alt3.gmail-smtp-in.l.google.com. gmail.com. 3599 în MX 40 alt4.gmail-smtp-in.l.google.com. gmail.com. 3599 în MX 5 gmail-smtp-in.l.google.com. ;; Timp de interogare: 20 msec ;; SERVER: 8.8.8.8 # 53 (8.8.8.8) ;; WHEN: Luni Aug 26 10: 13: 19 2016 ;; MSG SIZE rcvd: 150;

Putem observa că există mai multe înregistrări MX fiecare cu o valoare de nivel de preferințe diferită. Cu cât valoarea este mai mică, cu atât prioritatea este mai mare. Serverul de expediere va încerca mai întâi să livreze e-mail-ul către server cu cea mai mare prioritate și, în cazul nostru, este „gmail-smtp-in.l.google.com”

În continuare, trebuie să ne conectăm la serverul de poștă „gmail-smtp-in.l.google.com” din portul 25 (SMTP) pentru a confirma valabilitatea adresei de e-mail someuser@gmail.com. Pentru aceasta, putem folosi fie telnet, fie netcat. Ambele instrumente sunt disponibile din depozitele de software ale majorității distribuțiilor Linux.

$ nc gmail-smtp-in.l.google.com 25 # sau telent gmail-smtp-in.l.google.com 25 220 mx.google.com ESMTP a12si21630825itb.5 - gsmtp

Pentru a începe conversația HELO. Unele servere acceptă și EHLO în locul lui HELO.

HELO mydomain.com 250 mx.google.com la dispoziția dumneavoastră

Tip: mail de la: <nume@mydomain.com>

e-mail de la: <nume@mydomain.com> 250 2.1.0 OK v72si21823782itb.85 - gsmtp

Dacă serverul răspunde cu „250”, înseamnă că putem continua mai departe. Apoi, tastați: rcpt la: <someuser@gmail.com>

rcpt la: <someuser@gmail.com> 250 2.1.5 OK v72si21823782itb.85 - gsmtp

Răspunsul serverului ne va arăta dacă adresa de e-mail „someuser@gmail.com” este valabilă sau nu.  
Dacă primiți „250 OK” înseamnă că există adresa de e-mail.

Dacă primiți un răspuns de „550” așa cum se arată mai jos, înseamnă că acel cont de e-mail pe care ați încercat să îl accesați nu există.

rcpt la: <someuser2345@gmail.com> 550-5.1.1 Contul de e-mail pe care ați încercat să îl accesați nu există. Încercați 550-5.1.1 verificând de două ori adresa de e-mail a destinatarului pentru dactilografii sau spații inutile 550-5.1.1. Aflați mai multe la 550 5.1.1 https://support.google.com/mail/answer/6596 y18si12470464ioi.55 - gsmtp

**• Diferență dintre SSL și TLS**

Diferență dintre cele două este faptul că TLS a evoluat din tehnologia SSL și este o versiune mai sigură a SSL. De fapt, este TLS (și nu SSL) care este utilizat astăzi pentru a asigura comunicațiile online, dar chiar și atunci ne referim la acesta ca SSL, deoarece numele a rămas și este folosit mai frecvent.