

PROTOCOALE DE COMUNICAȚIE : LABORATOR 10

Protocoale pentru sistemul de poștă electronică (e-mail) în Internet

Responsabil: Alecsandru PATRASCU

Cuprins

Obiective	1
Prezentarea laboratorului	1
Anatomia mesajelor e-mail	2
Protocoalele SMTP, POP3, IMAP	3
SMTP	3
Exemplu	3
POP3	4
Exemplu	4
IMAP	4
Exemplu	4
Aplicație	5
Referințe	5

Obiective

În urma parcurgerii acestui laborator studentul va fi capabil să:

- Diferențieze și utilizeze două protocoale pentru citirea poștei electronice.
- Să folosească protocolul pentru trimiterea de mesaje și atasamente prin poșta electronică.
- Să scrie un client simplu de mail.

Prezentarea laboratorului

Primul mesaj e-mail a fost transmis în 1971 de un inginer pe nume Ray Tomlinson. Până la acea dată, puteau fi trimise mesaje doar în cadrul acelui calculator. Marea îmbunătățire introdusă de Tomlinson a fost posibilitatea de a trimite mesaje între calculatoare diferite din Internet, folosind semnul '@' pentru a desemna mașina spre care se trimite mesajul.

Azi se trimit miliarde de mesaje e-mail pe zi, și totuși multe din caracteristicile de atunci ale mesajelor au rămas. Vom vedea în continuare structura mesajelor email și modul în care sunt transmise acestea în Internet.

Anatomia mesajelor e-mail

Un mesaj e-mail a fost întotdeauna transmis în format plain-text. Chiar și prin adăugarea atașamentelor, mesajele de e-mail sunt trimise tot ca mesaje plain-text, prin folosirea unor mecanisme de codificare (uencode/udecode, MIME/BASE64).

Un mesaj este format dintr-o secțiune de header, urmată de o secțiune cu conținutul mesajului. Structura headerelor este descrisă în RFC 822, RFC 1521 și RFC 1806, ele având în general următoarea structură:

- Unul sau mai multe header *Received*, care indică ce cale a fost urmată de mesaj de la sursă până la destinație
- *Mime-Version*: versiunea mime folosită, 1.0 în general
- *Content-Type*: text/plain pentru mesaje text, multipart/mixed pentru mesaje cu atașamente
- *Subject*: - subiectul mesajului
- *Date*: - data și ora când a fost trimis mesajul
- *Message ID*: - un ID pentru mesaj, folosit pentru identificarea în mod unic a unui mesaj
- *From*: - numele și adresa de mail a expeditorului
- *To*: - numele și adresa de mail a destinatarului
- *Cc*: - carbon copy - alți destinatari
- Alte header introduse de clientul de mail folosit pentru a trimite mesajul

Conținutul mesajului este textul propriu-zis, pentru mesaje plain-text fără atașamente. Iată un exemplu de mesaj:

```
1 MIME-Version: 1.0
2 From: john@doe.edu
3 To: joe@doe.com
4 Subject: Food
5 Content-Type: text/plain
6
7 I'm hungry, Joe.
8
9 Cheers,
10 John.
```

Mesajele cu atașamente pot folosi una din următoarele tehnici pentru codificarea acestora:

- uuencode - la începuturile e-mail-ului, fișierele care se doreau trimise prin email trebuiau convertite în format text și invers prin folosirea utilitatelor numite uuencode/uudecode. Și în zilele de azi, unii clienți de mail adaugă atașamentele la sfârșitul mesajelor, codificându-le cu algoritmul folosit de uuencode.
- MIME / Base64 - aceasta tehnologie este cea recomandată pentru trimiterea de mesaje cu atașamente. Un mesaj cu atașamente codificate MIME arată în felul următor:

```
1 MIME-Version: 1.0
2 From: john@doe.com
3 To: joe@doe.com
4 Subject: Food
5 Content-Type: multipart/mixed; boundary=boundarystring
6 --boundarystring
7 Content-Type: text/plain
8
```

```
9 | I'm hungry, Joe.
10 |
11 |
12 | See the attachment for my pizza order.
13 |
14 |
15 | Cheers,
16 | John
17 |
18 |
19 | --boundarystring
20 | Content-Type: text/plain
21 | Content-Disposition: attachment; filename="textfile.txt"
22 | I want one large Sicilian-style pizza with
23 | mushrooms and black olives.
24 | --boundarystring
```

Se observa faptul ca părțile care compun mesajul sunt separate între ele printr-un "boundary string", specificat ca un parametru pentru header-ul Content-Type:. Fiecare parte poate avea la rândul ei propriile headere - în general acestea conțin tipul și numele fișierului conținut de partea respectivă. În cazul în care se trimit atașamente binare, acestea sunt codificate folosind schema numită Base64, descrisă în RFC 1521.

Protocoalele SMTP, POP3, IMAP

În terminologia folosită de sistemele de e-mail, există 3 actori în cadrul sistemului de e-mail. Aceștia pot fi situați pe 3 mașini diferite sau pot co-exista pe același computer:

1. *Mail User Agent (MUA)* - aplicația folosită de utilizator pentru a citi și trimite mesaje email (clientul de e-mail). El nu primește direct mesaje, acesta fiind rolul Mailbox Server-ului.
2. *Mailbox Server* - calculatorul/serverul care primește și stochează mesajele (server de e-mail).
3. *Mail Transfer Agent (MTA)* - aplicația care primește și retrimite mesajele spre un alt MTA sau spre un Mailbox Server ("router" de e-mail)

SMTP

SMTP (Simple Mail Transfer Protocol) este un protocol care se folosește pentru trimiterea mesajelor electronice (de la un client către un server). Acesta se folosește de portul 25

Exemplu

Vom folosi pe post de Mailbox Server un utilitar existent în Python:

```
1 | python -m smtpd -n -c DebuggingServer localhost:25
```

Atenție! Pentru a putea face bind pe un port mai mic de 1024, trebuie să aveți drepturi de sudo.

În continuare, pentru a trimite un mail se folosesc următoarele comenzi:

```
1 | telnet localhost 25
2 | HELO localhost
3 | MAIL FROM: user@localhost
```

```
4 RCPT TO: otheruser@localhost
5 DATA
6 Subject: Hello
7 To: otheruser@localhost
8 Test message <ENTER>
9 .<ENTER>
10 QUIT
```

POP3

POP3 (Post Office Protocol 3) este un protocol care se folosește pentru citirea mesajelor electronice (de la un server către un client). Clientul va interoga periodic serverul, va descărca mesajele și le va *sterge automat* de pe server. Comunicatia se realizează folosind portul 110.

Exemplu

```
1 telnet pop.example.com 110
2 USERNAME username
3 PASS password
4 LIST
5 RETR 1
6 QUIT
```

Pentru serverele de mail care folosesc conexiuni SSL, comanda de telnet (linia 1) se înlocuiește cu cea de mai jos. Trebuie să folosiți adresa furnizată de server (de ex: pop.mail.yahoo.com) și portul 995

```
1 openssl s_client -connect ADRESA_SERVER:995 -quiet
```

IMAP

IMAP (Internet Message Access Protocol) este un protocol care se folosește pentru citirea mesajelor electronice (de la un server către un client). Clientul va interoga periodic serverul și poate cere mesaje complete sau doar porțiuni (header, body), și *nu va șterge automat* mesajele de pe server. Comunicatia se realizează folosind portul 143.

Exemplu

```
1 telnet imap.example.com 143
2 a1 LOGIN username password
3 a2 LIST "" "*"
4 a3 EXAMINE Inbox
5 a4 FETCH 1 BODY[]
6 a5 LOGOUT
```

Pentru serverele de mail care folosesc conexiuni SSL, comanda de telnet (linia 1) se înlocuiește cu cea de mai jos. Trebuie să folosiți adresa furnizată de server (de ex: `imap.mail.yahoo.com`) și portul 993.

```
1 openssl s_client -connect ADRESA_SERVER:993 -quiet
```

Aplicație

Implementați un client simplu de SMTP, folosind socketi TCP. Porniți de la scheletul de cod atasat laboratorului.

Referințe

1. Anatomy of an email attachment – <http://www.uab.edu/it/email/index.html>
2. How E-Mail Works – <http://communication.howstuffworks.com/email.htm>
3. RFC 822 – Formatul mesajelor internet - <http://www.faqs.org/rfcs/rfc822.html>
4. RFC 821 – Protocolul SMTP - <http://www.faqs.org/rfcs/rfc821.html>
5. RFC 1725 – Protocolul POP3 - <http://www.faqs.org/rfcs/rfc1725.html>
6. RFC 1730 – Protocolul IMAP versiunea 4 - <http://www.faqs.org/rfcs/rfc1730.html>
7. RFC 1806 – <http://www.faqs.org/rfcs/rfc1806.html>