

Електронна поща

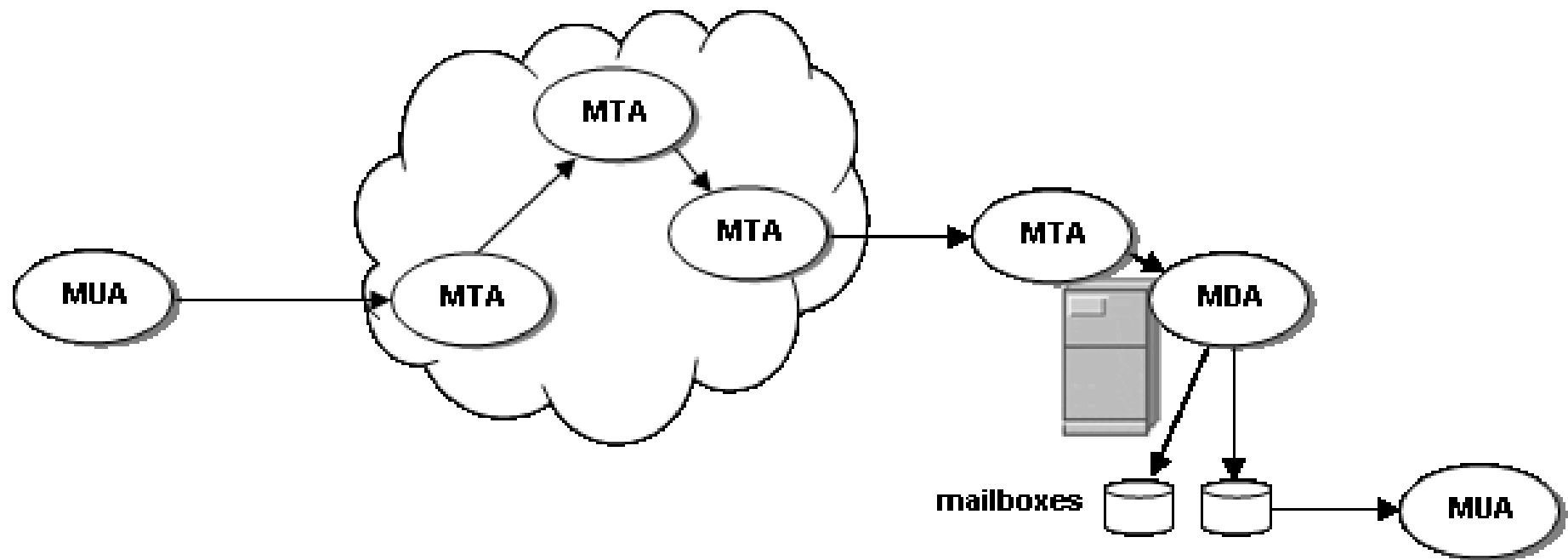
Електронна поща

- Начало - 1965г. с копиране съдържанието на файл от една машина на друга.
- Позволява трансфер само на текст.
- 1971г. в ARPANET се създава първото приложение за електронна поща –знакът @ отделя името на потребителя от името на машината
- 1977г.-RFC733-за унифициране на е-пощата
- 1982г.-RFC822 –за унифициране имената на домейни
- 1982г.-RFC821(RFC5321 от 2008г.)–SMTP, който заменя FTP за обмен на е-поща
- 1996г.-RFC2045-RFC2049-MIME (Multipurpose Internet Mail Extensions)

Характеристики

- Електронната поща използва DNS системата
- Всяко електронно съобщение включва заглавни части (хедъри) и тяло
- Използват се протоколи SMTP, POP и IMAP
- За предаване се използва порт TCP 25

Компоненти



Компоненти - MUA

Mail User Agent:

- Програма, която осъществява директния интерфейс с потребителя.
- Основните ѝ функции са свързани при предаване с формиране на e-mail съобщение и изпращането му към друга програма, която да осъществи неговото доставяне.
- При получаването на пощата, MUA осигурява прочитането на съобщенията от пощенската кутия на потребителя.

Примери: **mail** и **pine** под Linux, **Pegasus Mail** и **Outlook Express** под Windows

Компоненти - МТА

Mail Transport Agent:

- Специализирана програма, осъществяваща транспортирането на електронната поща между машините и нейната доставка до получателя.
- Тя получава подготвените съобщения от MUA, определя дестинацията и препраща писмата (relay) през други междинни МТА до машината на получателя – т.н. пощенски сървър (Mail server).

Примери: **sendmail**, **smail**, **Postfix** под Linux

Компоненти - MDA

Mail Delivery Agent:

- Осигурява доставянето на полученото от пощенския сървър писмо в пощенската кутия на конкретния получател.
- Поддържа и редица други функции като използване на псевдоними (aliases), пренасочване (forwarding) и списъци с получатели на електронна поща (mail lists).

Примери: **mail** под Linux

Последователност на изпращане на мейл

1. Създаване на електронната поща. MUA, изпълняващ се на компютъра на потребителя, формира електронното съобщение.
2. Изпращане на съобщението. MUA изпраща готовото писмо на съответния MTA. За мрежи, базирани на TCP/IP се използва протоколът SMTP (RFC2822), като и неговото разширение ESMTP.

Последователност на изпращане на мейл

3. Маршрутизиране на съобщението. Ако получателят е на същата машина, МТА извършва локално доставяне, с използване на услугите на локалния MDA. Когато получателят е на отдалечена машина, МТА осигурява предаването на съобщението до дестинацията. Обикновено, писмото преминава през различни хостове (relays), докато се получи от пощенския сървър на получателя. Като маршрутизиращи протоколи се използват SMTP и ESMTP.

Последователност на изпращане на мейл

4. Разпределяне на съобщенията. След получаване на електронната поща в крайния пощенски сървър, тя се обработва от съответния MDA и са записва в пощенската кутия на получателя.

5. Прочитане на съобщението. Потребителят прочита полученото писмо от своята пощенска кутия. Използват се основно два протокола: POP3 (Post Office Protocol ver.3), чрез който съдържанието на пощенската кутия на потребителя се прехвърля на локалния му компютър и IMAP4 (Internet Message Access Protocol ver.4), чрез който потребителят може освен това и да класифицира, изтрива и поддържа своите електронни писма динамично на пощенския сървър.

Адреси

Адрес на получателя:

идентификатор_на_локалния_потребител@описание_на_домейн

- идентификатор_на_локалния_потребител (изпращача или получателя)
 - потребителското име на съответната машина (login name),
 - реалното име във формат *ПървоИме.ПоследноИме*
 - псевдоним, който ще бъде преобразуван в потребителско име
 - списък от потребители.
- описание_на_домейн (пощенския сървър на изпращача или на получателя)
 - пълното квалифицирано име на домейн (FQDN) във вида *host.domain*.
 - не описва пътя до крайната машина.

Структура на мейл

Административна част

Envelope:

Адреси на източника и на
получателя

Mail header: темата на съобщението,
списък от множество получатели,
времето на изпращане на
съобщението и др

Празен ред

Body

```
Received: from nobody by miami.hostmeta.com.br with local (Exim 4.82)
(envelope-from )
id 1Nov0w-0005Cr-3b
for myemail@mydomain.dom; Mon, 26 May 2014 10:32:30 -0300
To: myemail@mydomain.dom
Subject: =?iso-8859-1?Q?Informativo_urg=EAnte_de_atualiza=E7=E3o?=
Date: Mon, 26 May 2014 10:32:30 -0300
From: =?iso-8859-1?Q?ComunicadosIta=FA?=
Message-ID:
X-Priority: 3
X-Mailer: PHPMailer [version 1.73]
MIME-Version: 1.0
Content-Transfer-Encoding: 8bit
Content-Type: text/html; charset="iso-8859-1"
X-AntiAbuse: This header was added to track abuse,
please include it with any abuse report
X-AntiAbuse: Primary Hostname - miami.hostmeta.com.br
X-AntiAbuse: Original Domain - gmail.com
X-AntiAbuse: Originator/Caller UID/GID - [99 99] / [47 12]
X-AntiAbuse: Sender Address Domain - miami.hostmeta.com.br
X-Get-Message-Sender-Via: miami.hostmeta.com.br: uid via
acl_c_vhost_owner from authenticated_id: nobody from /only user
confirmed/virtual account not confirmed
```

Полета

- Подател
 - From:**-Адрес на автора на писмото;
 - Sender:**-Потребителския софтуер, използван за съставяне на писмото;
 - Reply-To:**-Адрес на отговора, често използван от мейллисти;
- Получатели
 - To:**-Основен списък от получатели, разделени с „“;
 - Сс:**-Списък с други получатели, към които ще се изпратят копия;
 - Всс:**-Списък с други получатели, към които ще се изпратят копия на писмото, без обаче те да бъдат информирани за останалите получатели;
- По време на изпращането
 - Date:**-Датата на изпращане на писмото;
 - Message-Id:**-Уникален идентификатор на съобщението, генериран от МТА на подателя;
 - Subject:**-Темата на съобщението;

Полета

- По време на маршрутизирането
 - Received:**-Всеки междинен МТА,които получава писмото, добавя в това поле името на своята машина, датата и идентификатора на съобщението, откъде е получено и името на софтуера, който го е транспортирал;
 - Status:**-Индикация за статуса на всяко получено

```
Received: from host2 by host3
Received: from host1 by host2
Received: from host0 by host1
Received: by host0          ← Sender

host3 ← host2 ← host1 ← host0
```

Маршрутизиране

Received: from **CGNET.COM** by **Arizona.EDU** (PMDF V4.3-9 #2381)
id <01HGUMM9OTUO9AR7DY@Arizona.EDU>; Thu, 08 Sep 1994
00:39:13 -0700 (MST) **host2**

Received: from **faop.cgnet.com** by **CGNET.COM** (PMDF V4.3-9 #7702)
id <01HGUMN7N4S000370I@CGNET.COM>; Thu, 08 Sep 1994 00:40:08
-0700 (PDT)

host1 Received: from **msmail.fao.org** (191.0.1.130)
by **FAOVMS.CGNET.COM** (PMDF V4.3-8 #3703)
id <01HGV4ZD1XTC8WW39N@FAOVMS.CGNET.COM>; Thu, 08 Sep 1994
09:25:10 +0200

Received: by **msmail.fao.org** with Microsoft Mail id
<2E79C6AC@**msmail.fao.org**>;
Thu, 08 Sep 94 09:24:12 +02 **host0**

MIME (Multipurpose Internet Mail Extension)

- Изпращане на разнообразна информация през Интернет;
- Кодиране и прикачане на двоични (не-ASCII) данни към съобщенията;
- Използване на много типове данни в един и същ документ;
- Средства за идентифициране на типа на съдържанието;
- Неограничена дължина на съобщението

Дефиниране на данни

MIME дефинира типа данни посредством 3 описателни части: тип, подтип и опционни параметри на съдържанието, които се указват в полето Content-type, съдържащо се в заглавието на съобщението.

Content-type: type/subtype [;parameters]

Типове данни

Тип	Значение	Подтип
text	текст	plain: не включва форматираща информация
audio	звук	basic
image	статично изображение	формат на изображението, например, jpg, gif, tiff
video	анимирано изображение	формат на изображението, например, mpeg, quicktime
application	данни за приложение	octet-stream, postscript
message	данните представляват пакетизирано e-mail съобщение	rfc822, partial, external-body
multipart	съобщение с множество типове данни. Подтипът определя как тези части се интерпретират	mixed, alternative, digest, parallel

Кодиране

- **7bit** – Използва се за 7-битови US-ASCII символи без прекодиране;
- **8bit** – Използва се за 8-битови текстови данни. Може да се използва единствено ако и подателя и получателя МТА поддържат 8-битово MIME прекодиране;
- **Quoted-printable** – Прилага се към ASCII текст, в който само малка част от символите са не-ASCII, например, скандинавските символи в стандарта ISO8859-1. Процесът на прекодиране игнорира ASCII символите и прекодира единствено останалите;
- **Base64** – Прилага се при двоични данни. Алгоритъмът прекодира множество от 3 октета от данни в четири 6-байтови ASCII символи като въвежда увеличение на обема на съобщението от около 33%.

Примери

Съдържание в обикновен ASCII текст:

```
Mime-Version: 1.0  
Content-type: text/plain; charset="us-ascii"  
...
```

Съдържание хипертекстов документ, символно множество iso-8859-1 и начин на прекодиране quoted-printable:

```
Mime-Version: 1.0  
Content-type: text/html; charset="iso-8859-1"  
Content-Transfer-Encoding: quoted-printable  
...
```

Примери

Прикачане на документ в pdf формат и начина на прекодиране – base64:

```
Mime-Version: 1.0
```

```
Content-type: application/pdf; name="manual.pdf"
```

```
Content-Transfer-Encoding: base64
```

```
...
```

Разширения

Една от характерните възможности на MIME е включването на няколко различни типове данни в едно съобщение – multipart разширение. Отделните части се разграничават с последователност от символи, които трябва да се укажат в полето **Content-type**. Тези ограничители се разполагат и в началото и в края на тялото на електронното съобщение.

В зависимост от необходимия начин на интерпретиране, отделните части могат да бъдат от един от следните подтипове:

- **mixed** – Всяка част представя самостоятелна информация, като отделните части се разглеждат в определената последователност;
- **alternative** – Всяка част се явява различно представяне на една и съща информация. Системите трябва да изберат “най-добрия” тип за извеждане на базата на локалните променливи на средата;
- **parallel** – Всяка част представя самостоятелна информация, но редът на следване не е от значение. Най-честото приложение на този тип е едновременното изобразяване на съдържанието на всичките части от съответния софтуер или хардуер;
- **digest** – Всяка част е RFC2822 съобщение, използвано от пощенски списък.

Пример

```
Content-type: multipart/mixed; boundary="frontier"  
MIME-version: 1.0
```

```
--frontier
```

```
Content-type: text/plain
```

```
This is the body of the message.
```

```
--frontier
```

```
Content-type: application/octet-stream
```

```
Content-transfer-encoding: base64
```

```
gajwO4+n2Fy4FV3V7zD9awd7uG8/TITP/vIocxXnnf/5mjgQj cipBUL1  
b3uyLwAVtBLOP4nV  
LdIAhSz1ZnyLAF8na0n7g6OSeej7aqI13NIXCfxDsPsY6NQjSvV77j4h  
WEj1F/ag1S6ghfju  
FgRr+OX8QZMI1OmR4rUJUS7xgoknalqj3HJvaOpeb3CF1NI9VGZYz6H6  
zuQBOWZzNB8glwpC  
--frontier--
```

Маршрутизиране на пощата

След формиране на писмото от MUA и изпращането му към MTA, MTA изгражда списък със сайтовете на получателите и поставя по едно писмо в т.н. плик за всеки сайт-получател.

- За всеки сайт MTA трябва да определи метода за маршрутизация на базата на адреса на получателя.
- Първо се определя хоста, към който трябва да се изпрати съобщението, след което то се доставя директно към този хост.
- За анонсирането на мейл услуга, в локалната DNS система на даден сайт за пощенския сървър трябва да има специален ресурсен запис – MX (Mail Exchanger).

Маршрутизиране на пощата

Всеки MX запис съдържа специален параметър – предпочитание (preference), представляващ цяло число. Ако съществуват няколко пощенски прехвърлящи машини (mail exchangers) към даден хост, то транспортният агент ще се опита да предаде съобщението през прехвърлящ сървър с най-малка стойност на предпочитанието. В случай на неуспех, МТА ще избере сървър с по-висока стойност.

<code>games.corp.</code>	<code>IN</code>	<code>MX</code>	<code>5</code>	<code>mail.games.corp.</code>
<code>games.corp.</code>	<code>IN</code>	<code>MX</code>	<code>10</code>	<code>mail2.games.corp.</code>

Протокол SMTP (Simple Mail Transport Protocol)

- Протоколът SMTP позволява клиентите да доставят електронната поща до съответните пощенски сървъри.
- Сървърите използват също SMTP, за да препратят съобщенията между други пощенски сървъри до окончателното доставяне на писмото в пощенската кутия на получателя.
- Стартиране на SMTP сесия се извършва винаги от страна на MUA, докато транспортният агент очаква получаването на SMTP съобщения на порт 25.

Протокол ESMTP

- От ноември 1995
- RFC1869
- Предава 8-битови ASCII символи (RFC 6152)
- Поддържа MIME
- Има индикация за максималната големина на съобщението (RFC 1870)
- Има автентикация(RFC 4954)

Команди на SMTP

- Командите дефинират предаването на писмата или функциите, изисквани от потребителя.
- Някои команди изискват и съответен аргумент.
- Отговорът на дадена команда е трицифрен код.
- Командите са символни низове, завършващи с <CRLF>.
- Ако командата има параметри, те се разделят от името и с <Space>.

command arguments <CRLF>

Основни команди

HELO (HELLO)

Това е първата команда в сесията. Използва се за идентифициране на SMTP изпращача от SMTP получателя. Аргументът съдържа името на домейна на SMTP изпращача. При получаване на отговор **OK** това е индикация, че SMTP изпращачът и получателят са в началния стадий на обмен.

EHLO (HELLO)

Алтернативна команда за стартиране на диалог по разширения протокол ESMTP.

Основни команди

MAIL FROM

Указва изпращача на писмото. Като аргумент се изисква e-mail адреса на изпращача. Всеки върнат e-mail ще бъде изпратен към този адрес. Командата инициира мейл транзакция в която данни се доставят до една или повече пощенски кутии. Аргументът може да съдържа и `reverse-path` - опционен списък от хостове.

RCPT TO (RECIPIENT TO)

Указва получателя на писмото. Като аргумент се задава e-mail адреса на получателя. Множество получатели могат да се зададат чрез няколко такива команди. Аргументът може да съдържа и **`forward-path`** – опционен списък от хостове. Когато има списък с хостове, това е път на източника и указва, че мейлът трябва да се релейва към следващ хост от списъка. Ако SMTP-получателят не реализира релейване се изпраща код за грешка 550.

Основни команди

Когато мейлът се релейва, релей хоста трябва да се премахне от началото на forward-path и да се постави в началото на reverse-path. Когато мейлът достигне до крайната дестинация (forward-path ще съдържа само пощенската кутия на получателя) SMTP-получателят поставя съобщението в пощенската кутия. Например, на хост А е получен мейл с аргументи:

FROM:<USERX@HOSTY.ARPA>

TO:<@HOSTA.ARPA,@HOSTB.ARPA:USERC@HOSTD.ARPA>

Мейлът ще бъде релейван към хост В с аргументи:

FROM:<@HOSTA.ARPA:USERX@HOSTY.ARPA>

TO:<@HOSTB.ARPA:USERC@HOSTD.ARPA>

Основни команди

- **DATA**
- Задава данните, които ще формират тялото на писмото. Краят на данните се указва със символа ‘.’ на отделен ред, т.е. последователността от символите ‘<CRLF>.<CRLF>’. Когато SMTP-получател получи съобщение за релейване или за крайна доставка, той вмъква в началото на данните ред с времеви маркер. Той включва идентификация на хоста-подател, хоста-получател и дата/време на получаване на съобщението. Релейваните мейли ще имат множество такива редове.
-
- **QUIT**
- Указва завършване на диалога между изпращача и получателя. Получателят трябва да изпрати отговор ОК и тогава да затвори комуникационния канал. Получателят не трябва да затваря канала докато не получи и отговори на QUIT (дори и да има грешка). Подателят не трябва да затваря канала докато не изпрати QUIT и получи отговор (дори и да има отговор за грешка от предишна команда).

Отговори

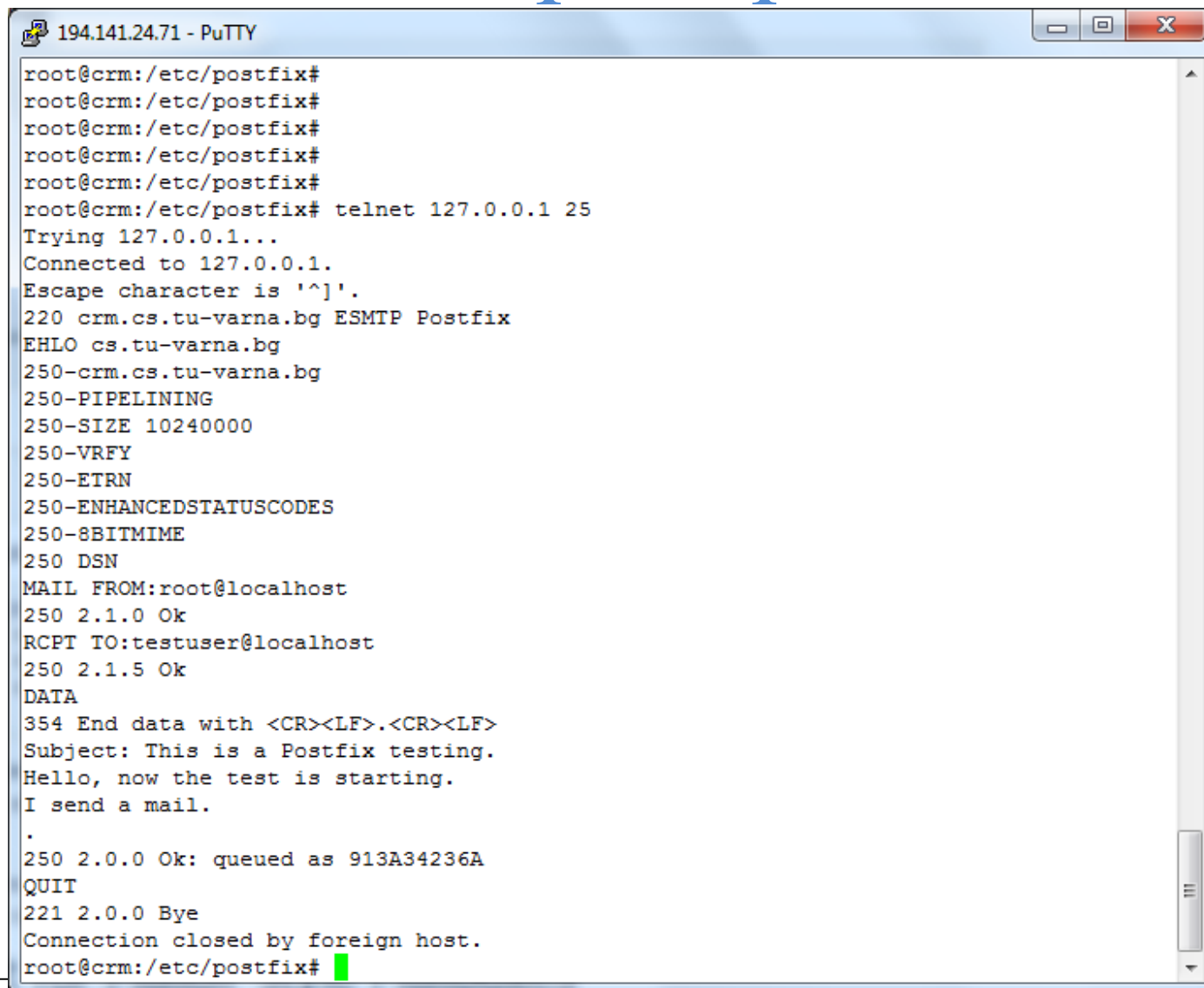
- 1-та цифра –успех (1,2,3), провал (5) или незавършен (4)
- 2-та цифра –категорията на съобщението за грешка (0-синтактична; 2-проблем с връзката; 5-проблем с мейла)
- 3-та цифра –специфичното съобщение в тази категория
- Примери:

250 –изисканото действие е изпълнено

550 –пощенската кутия на получателя не може да бъде открита

452 –препълнена е паметта на пощенския сървър

Пример



A screenshot of a PuTTY terminal window titled "194.141.24.71 - PuTTY". The terminal shows a series of commands and responses from a Postfix mail server. The user is in the directory `/etc/postfix` and runs `telnet 127.0.0.1 25`. The connection is successful, and the server responds with standard SMTP banners and capabilities. The user then sends an email with the subject "This is a Postfix testing. Hello, now the test is starting. I send a mail." and the email is successfully queued.

```
root@crm:/etc/postfix#  
root@crm:/etc/postfix#  
root@crm:/etc/postfix#  
root@crm:/etc/postfix#  
root@crm:/etc/postfix#  
root@crm:/etc/postfix# telnet 127.0.0.1 25  
Trying 127.0.0.1...  
Connected to 127.0.0.1.  
Escape character is '^]'.  
220 crm.cs.tu-varna.bg ESMTP Postfix  
EHLO cs.tu-varna.bg  
250-crm.cs.tu-varna.bg  
250-PIPELINING  
250-SIZE 10240000  
250-VERFY  
250-ETRN  
250-ENHANCEDSTATUSCODES  
250-8BITMIME  
250 DSN  
MAIL FROM:root@localhost  
250 2.1.0 Ok  
RCPT TO:testuser@localhost  
250 2.1.5 Ok  
DATA  
354 End data with <CR><LF>.<CR><LF>  
Subject: This is a Postfix testing.  
Hello, now the test is starting.  
I send a mail.  
.  
250 2.0.0 Ok: queued as 913A34236A  
QUIT  
221 2.0.0 Bye  
Connection closed by foreign host.  
root@crm:/etc/postfix#
```

Получаване на поща

Три базови модела:

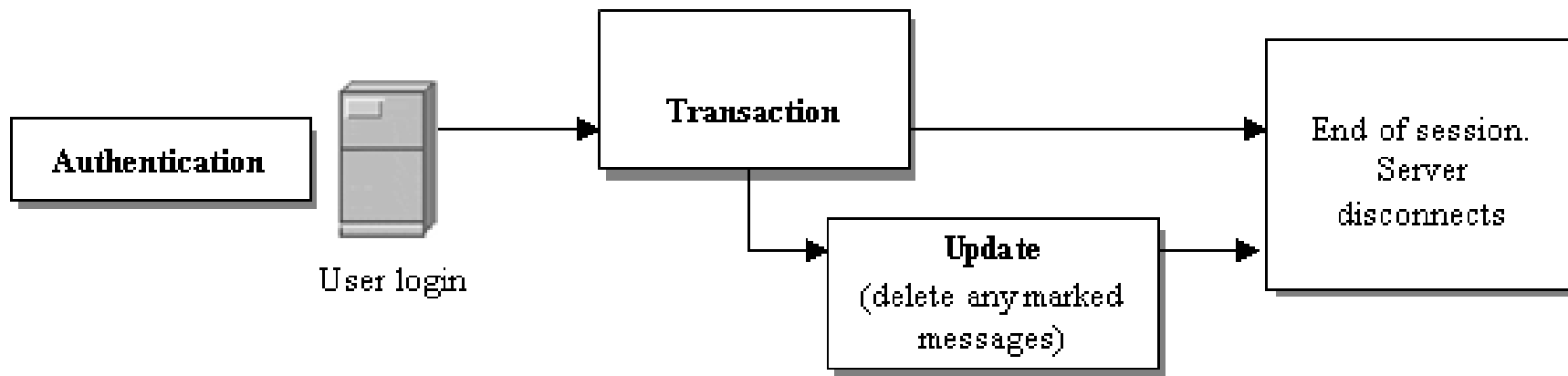
- **Off-line** (POP3, IMAP4)—изтеглят се съобщенията на локалния компютър
- **On-line** (IMAP4)—съобщенията се манипулират директно на пощенския сървър без необходимостта от тяхното изтегляне на локалния компютър.
- **Disconnected use** (IMAP4)—свързва се със сървъра, синхронизира съобщенията като изтегля в локалния кеш новите и приключва връзката

Протокол POP3

RFC1939-POP3 (Post Office Protocol)

- Използва се от пощенски клиенти за получаването на електронната поща от пощенските кутии на потребителя.
- Текстово базиран.
- Отговорите от сървъра съдържат обозначение за успешно изпълнение на командата (+OK) или за грешка (-ERR).
- Клиентите се свързват към пощенските сървъри по порт 110.

Фази на обмен



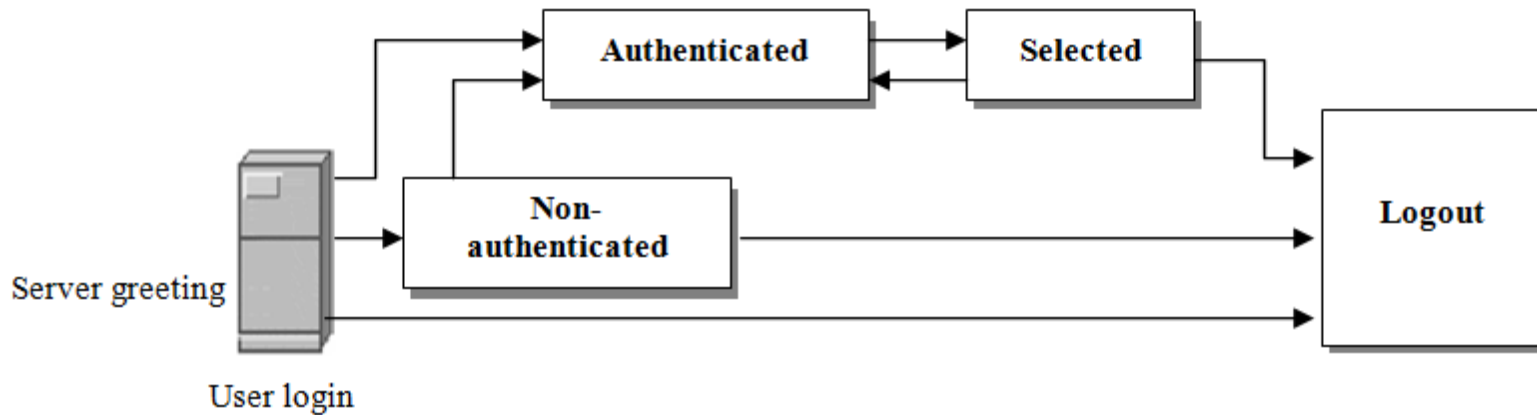
- **Authentication state.** Клиентът се автентичира като изпраща потребителско име и парола. По време на процеса на автентикация, пощенската кутия на потребителя се В даден момент само един потребител може да се свърже към определена пощенска кутия;
- **Transaction state.** През тази фаза клиентът изпраща последователност от команди, които сървърът обработва и на които отговаря съобразно протокола;
- **Update state.** Заключителната фаза при комуникацията между клиента и сървъра, при която връзката се затваря. След затварянето на връзката, съдържанието на пощенската кутия се обновява за да отрази промените, които клиентът е направил по време на последната сесия.

Протокол IMAP4

RFC2060-IMAP4 (Internet Message Access Protocol)

- Използва се на порт 143
- Работи с текстови команди , изпращани от клиента и връщане на данни и статуса на изпълнението им от сървъра
- Характеристики:
 - ефективно управление на папки - създаване, преименуване и изтриване
 - разширено търсене на съобщения по техния размер, заглавни част и податели.
 - отделяне на прикачените файлове от заглавната част и текста на съобщението –полезно при MIME съобщения с много части.
 - Паролата на потребителя за автентикация не се изпраща в чист текст. Същевременно, сървърът периодично изпраща проверяващи съобщения (challenges) за удостоверяване идентичността на свързания потребител.

Фази на обмен



- **Non-authenticated** - При стартиране на връзката клиентът трябва да предаде своите креденциали (име, парола) преди повечето команди да бъдат разрешени. Ако конекцията е била предварително автентизирана, директно се преминава към следващото състояние;
- **Authenticated** - След като клиентът се е автентизирал, сесията преминава в това състояние. Клиентът първо трябва да избере пощенска кутия за достъп, преди командите за обработка на съобщенията да бъдат разрешени.
- **Selected** - Сесията преминава в това състояние когато успешно е избрана пощенска кутия;
- **Logout** - Конекцията се прекратява от сървъра след съответна заявка от страна на клиента или едностранно от сървъра.

Въпроси ?