Протоколы, сервисы и услуги
в Интернет и IP-сетях

Тема № 13
Электронная почта

доц. каф. СС и ПД, к.т.н. С. С. Владимиров

2017 г.

Электронная почта. Формат электронного адреса

Электронная почта (Electronic mail, e-mail)

Технология и предоставляемые ею услуги по пересылке и получению электронных сообщений по компьютерной сети.

Электронная почта по составу элементов и принципу работы практически повторяет систему обычной почты, заимствуя, в частности, термины (почта, письмо, конверт, вложение, ящик, доставка и другие).

Электронная почта в Internet использует *маршрутно-независимую адресацию*. Это значит, что адрес пользователя остается неизменным независимо от того, откуда посылается сообщение. Такая адресация очень удобна для пользователей, но усложняет процесс доставки сообщения, так как определение маршрута доставки полностью ложится на программное обеспечение электронной почты.
Допускается также указание маршрута сообщения в адресе получателя, но такие маршрутно-зависимые адреса используются редко, обычно, в отладочных целях. В общем случае их использование не имеет смысла.

Формат электронного адреса

Формат электронного адреса подробно описан в RFC 2822 . В общем виде он имеет следующий формат:

```
имя_пользователя@почтовый_домен
```

- `имя_пользователя` – идентификатор пользователя, уникальный в пределах одного почтового домена;
- `@` – символ-разделитель (коммерческое at);
- `почтовый_домен` – уникальный идентификатор почтовой системы.

Формат электронного адреса

Имя пользователя

Имя пользователя может состоять из цифр, латинских букв и символов:

```
! # $ % & ’’ * + - / = ? ^ _ ’ { | } ~
```

Оно может состоять из нескольких полей, разделенных точкой. Если имя пользователя содержит символы, отличные от перечисленных, его следует заключать в кавычки.
На практике имена пользователей обычно состоят только из цифр, латинских букв, символов «-», «_» и точки, которая интерпретируется не как разделитель полей, а как часть имени пользователя. Использование других символов может привести к неоднозначным интерпретациям, потому не желательно.

Почтовый домен

Имя почтового домена имеет тот же формат, какой используется в доменных именах Internet. Описан в RFC 1034.

Комментарии

Кроме значимой части, используемой при маршрутизации сообщения, адрес может содержать комментарии в виде произвольных текстовых строк до и после значимой части, отделенных от нее угловыми скобками.

```
комментарий < имя_пользователя@почтовый_домен > комментарий
```

Информация по обе стороны угловых скобок при доставке сообщения игнорируется.
Адрес электронной почты не обязательно указывает непосредственно на существующий почтовый ящик. Это также может быть почтовый псевдоним, указывающий на другой адрес, или адрес списка рассылки, указывающий на множество других адресов, или адрес, на который приходят сообщения, поступающие на обработку специальной программой, и т. д.
Некоторые системы допускают также использование одного лишь имени пользователя в качестве электронного адреса, если получатель зарегистрирован в том же почтовом домене, из которого посылается сообщение.

Формат сообщения электронной почты

Формат сообщения, передаваемого по электронной почте, описан в RFC 2822 . Оно состоит из трех частей:

- *Конверт* (envelope), содержащий адреса отправителя и получателей сообщения, эта информация используется только при пересылке сообщения по протоколу SMTP , получателю она недоступна.
- *Заголовок* (header), содержащий служебную информацию, формируемую программами, участвующими в передаче сообщения, такую как адреса отправителя и получателей, которые могут отличаться от используемых в конверте, тему сообщения, время отправки, сведения о пересылке и об используемых для создания сообщения программах и т. д. Заголовок завершается пустой строкой.
- *Тело* (body), содержащее само сообщение, созданное отправителем и подлежащее доставке получателю.

Таким образом, сообщение доставляется получателю в виде заголовка и отделенного от него пустой строкой тела.
Заголовок состоит из полей: текстовых строк, состоящих из имени поля: слова, заканчивающегося двоеточием, и содержимого поля.
В заголовке допускается использование только символов в кодировке US-ASCII. Другие символы должны быть закодированы таким образом, чтобы полученная кодовая последовательность содержала только символы кодировки US-ASCII. Это правило нередко нарушается, например, тема сообщения записывается в заголовке сообщения на русском языке без перекодирования. Этого следует избегать, так как на приемном конце не будет известна используемая кодировка русских букв, а значит, полученная последовательность может быть интерпретирована неправильно. Этого не произойдет, если текст будет закодирован в соответствии с RFC 2047 .
Длинные поля заголовка разбиваются на несколько строк, при этом каждая строка, продолжающая предыдущую, начинается с пробельного символа. Заголовок обычно показывается не полностью. Получатель видит только некоторые поля: адреса отправителя и получателей, время отправки и тему сообщения.
Тело сообщения, если это не просто текст, записанный латинскими буквами, должно быть закодировано в соответствии со спецификацией MIME, как описано в RFC 2045. На приемной стороне оно при необходимости декодируется и преобразуется в понятный пользователю вид.

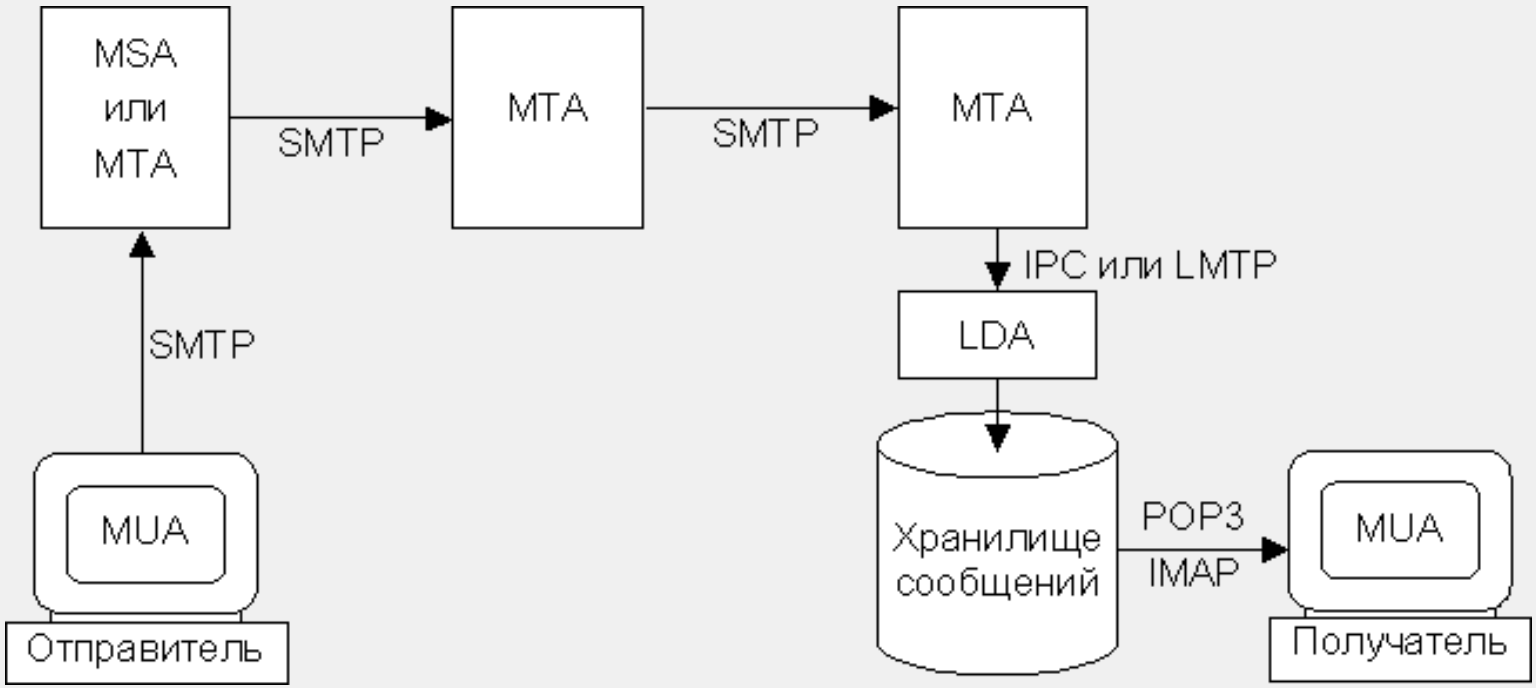
Пример сообщения электронной почты

Пример сообщения

```
Return-Path: <gmailaddr@gmail.com>
Received: from blacktower ([91.101.175.183])
        by mx.google.com with ESMTPSA id j2sm10233782lag.12.2014.03.23.03.44.05
        for <gmailaddr@gmail.com>
        (version=TLSv1 cipher=RC4-SHA bits=128/128);
        Sun, 23 Mar 2014 03:44:07 -0700 (PDT)
Message-ID: <532ebaf7.421a980a.4f0a.ffffc585@mx.google.com>
Received: by blacktower (sSMTP sendmail emulation); Sun, 23 Mar 2014 14:44:04 +0400
From: root <gmailaddr@gmail.com>
Date: Sun, 23 Mar 2014 14:44:04 +0400
To: root
Subject: Anacron job 'cron.daily' on blacktower
Content-Type: text/plain; charset=US-ASCII

/etc/cron.daily/logrotate:
ERROR Unable to contact server. Is it running?
error: error running non-shared postrotate script for /var/log/fail2ban.log of
'/var/log/fail2ban.log
'run-parts: /etc/cron.daily/logrotate exited with return code 1
```

Структура электронной почты в Internet



Компоненты электронной почты

MUA Mail User Agent – пользовательский агент, или клиентская почтовая программа.

MTA Mail Transfer Agent – транспортный агент, или почтовый сервер.

LDA Local Delivery Agent – агент локальной доставки.

MSA Message Submission Agent – агент подачи сообщения.

Mail User Agent (MUA)

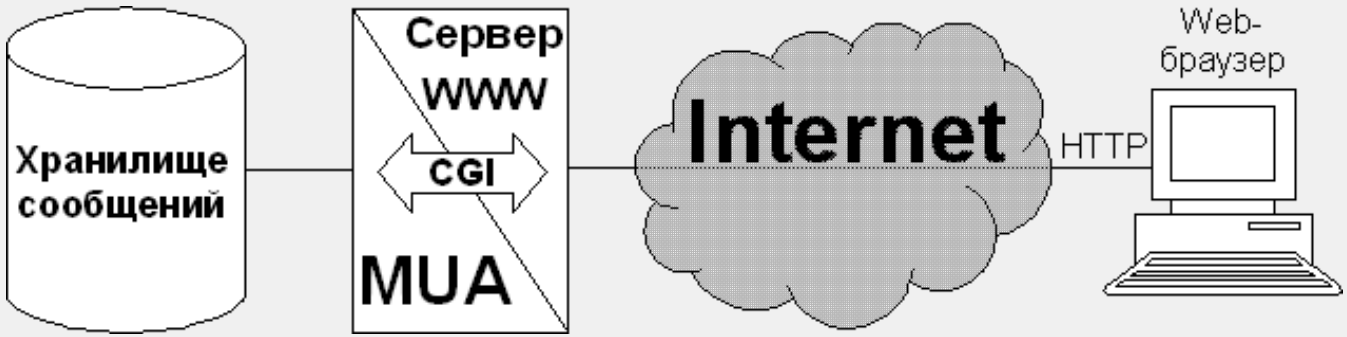
MUA предназначен для подготовки, отправки, получения и просмотра электронных писем. Это программа, установленная на компьютере пользователя. Задача электронной почты, по сути дела, сводится к тому, чтобы доставить сообщение от MUA отправителя на MUA получателя.
Подготовка к отправке заключается в приведении сообщения к принятому в Internet формату, описанному в RFC 2822.
MUA отправителя должен сформировать заголовок сообщения, а также закодировать и оформить его тело в соответствии со стандартом, чтобы MUA принимающей стороны смог правильно интерпретировать и представить как текст, так и вложения письма.
Так как MUA обычно устанавливается на машине пользователя, он, как правило, запускается только на время работы пользователя, а компьютер, на котором запущен MUA, может не иметь постоянного подключения к Internet. Поэтому MUA не может выступать в качестве сервера – он может быть только инициатором соединения, то есть клиентом.
MUA посылает сообщения по протоколу SMTP через MSA или MTA, используемый для отправки почты.
Входящие письма MUA забирает из хранилища сообщений по протоколу, предназначенному для получения почты. Как правило для этой цели используется один из двух протоколов:

1. Post Office Protocol Version 3 (POP3) – протокол почтового отделения, версия 3, описанный в RFC 1939, позволяет просматривать сообщения в почтовом ящике, забирать и удалять их.
2. Internet Message Access Protocol (IMAP) – протокол доступа к сообщениям, описанный в RFC 3501, обладает более широкими возможностями манипулирования почтовыми ящиками, чем POP3, в частности он позволяет работать с несколькими ящиками одновременно, не только считывать и удалять, но и создавать и исправлять сообщения.

Возможны и другие способы получения почты. Например, использование локальной доставки, если хранилище сообщений доступно MUA по локальной сети.

Mail User Agent (MUA). Web-mail

Довольно большое распространение получили агенты пользователя, использующие интерфейс CGI для доступа оконечного пользователя к его почтовому ящику по протоколу HTTP или более безопасному HTTPS при помощи web-браузера. Такую реализацию MUA часто называют web-mail.



Пользовательский интерфейс реализуется с помощью технологий WWW . Функции MUA выполняет приложение, взаимодействующее с web-сервером при помощи интерфейса CGI. MUA получает доступ к хранилищу сообщений по протоколам POP3 или IMAP или путем непосредственного обращения – MUA при такой реализации может быть включен в ту же локальную сеть, что и хранилище сообщений, они даже могут быть запущены на одной и той же машине.

Преимущество web-mail перед MUA, установленным на компьютере пользователя, это возможность работать со своей почтой с любого компьютера, подключенного к Internet, без предварительной настройки и без инсталляции программного обеспечения. Недостаток web-mail заключается в том, что пользователю для работы с почтой необходим постоянный доступ к Internet, так как каждый запрос выполняется не на пользовательской машине, а на web-сервере, и должен быть передан по сети.

Mail Transfer Agent (MTA)

MTA представляют собой узлы, через которые передаются электронные сообщения. Письмо, сформированное MUA, достигает хранилище сообщений, содержащее почтовый ящик получателя, проходя через один или несколько MTA, последний из которых передает письмо агенту локальной доставки (LDA).
Как правило, MTA должны быть доступны круглосуточно и постоянно ожидать подключения по протоколу SMTP. Иными словами, каждый MTA в Internet включает в себя сервер SMTP. Обмен данными между MTA происходит по этому протоколу. MTA, отправляющий почту, инициирует соединение и выступает в качестве клиента, MTA, принимающий почту, является сервером.
На MTA также возлагается разбор адресов получателей, раскрытие списков рассылки и почтовых псевдонимов и определение маршрута сообщения на основании анализа адресов получателей и записей MX, получаемых от сервера DNS.
MTA должен проверять соответствие действительности идентификационных данных получаемых им от встречного MTA. Следует проверять соответствие доменного имени, которое клиент сообщает в приветствии, его адресу IP. Также нужно удостовериться в существовании почтового домена, указанного в почтовом адресе отправителя. Если в доменной части адреса получателя указан почтовый домен, обслуживаемый данным MTA, то следует проверить, зарегистрирован ли в этом домене указанный адресат.
В целях предотвращения анонимных рассылок спама, RFC 2505 рекомендует принимать почту только при выполнении хотя бы одного из следующих условий:

- адрес IP клиента входит в список адресов клиентов, обслуживаемых данным MTA;
- получатель сообщения зарегистрирован в почтовом домене, обслуживаемом данным MTA;
- клиент прошел процедуру аутентификации.

Если не выполнено ни одно из названных условий, MTA должен отказать в приеме почты. MTA, принимающий почту, не отвечающую перечисленным требованиям, может быть внесен в списки серверов, не препятствующих распространению спама. В этом случае многие почтовые системы будут отказываться принимать от него почту.

Mail Transfer Agent (MTA)

MTA может производить обработку проходящих через него сообщений: проверку на наличие вирусов, фильтрацию спама и пр.
Каждый MTA, через который проходит почтовое сообщение, добавляет к его заголовку информацию о том, когда и откуда пришло это сообщение, а также результаты произведенных проверок.
В случае невозможности немедленной доставки сообщения, оно помещается в очередь. MTA регулярно предпринимает новые попытки отправить сообщения из очереди. Если это не удается за определенный срок, обычно за четыре часа, то отправителю посылается предупреждение о задержке доставки. Но сообщение остается в очереди, и попытки его отправить продолжаются. Если в течение длительного времени, обычно пяти дней, сообщение так и не удается доставить, оно удаляется из очереди, а отправителю посылается сообщение о невозможности доставки письма.
MTA могут также выполнять и другие функции, в зависимости от используемого программного обеспечения.
Основные требования к MTA и к MUA описаны в RFC 1123 и уточнены в RFC 2821 и в RFC 2822.
Существует множество разнообразных программных реализаций MTA. Старейшей из них и до сих пор одной из наиболее популярных является программа sendmail, разработанная в начале восьмидесятых годов Эриком Оллмэном, тогда еще студентом Калифорнийского университета в Беркли. Эта программа многократно дорабатывалась и стала фактически стандартом для этого типа программного обеспечения. Она продолжает совершенствоваться и по сей день. Доступна как свободно распространяемая версия для операционных систем, совместимых с UNIX, так и коммерческая версия.
Позже появились и другие программные продукты, реализующие функции MTA для различных операционных систем: Postfix, smail, qmail, exim, ZMailer и многие другие.

Message Submission Agent (MSA)

MTA, через которые проходит сообщение, добавляют некоторые строки в его заголовок. Однако информацию, уже содержащуюся в сообщении, MTA не изменяют, хотя необходимость в этом может возникнуть. Заголовок сообщения, полученного от MUA, может быть неправильно оформлен, например, там может быть не определено полностью имя домена, ошибочно указано время или дата. Может возникнуть необходимость в корректировке адреса отправителя, если в почтовой сети предприятия используется адресация, отличная от принятой в Internet. Например, если допускается использование адресов без указания почтового домена для пользователей, зарегистрированных в почтовой системе предприятия.

Функции корректировки заголовка сообщения можно возложить на MTA, принимающий почту от агентов пользователя, но, если поток почты велик, имеет смысл использовать для этого специальный процесс – MSA. Таким образом, можно сказать, что MSA это разновидность MTA, занимающаяся предварительной обработкой исходящей почты. Подробнее задачи и особенности реализации MSA описаны в RFC 2476.

Чтобы различать MTA и MSA, рекомендуется запускать MSA не на порту 25, предназначенном для MTA, а на другом порту TCP, либо использовать порт 25 на сервере, где не запущен MTA.

Local Delivery Agent (LDA)

Последний MTA на пути следования электронного почтового сообщения должен передать его агенту локальной доставки. Обычно LDA расположен на одной машине с MTA и представляет собой программу, которая вызывается агентом передачи сообщения при поступлении новых сообщений. В этом случае для взаимодействия между MTA и LDA используются механизмы межпроцессного взаимодействия (IPC). В некоторых случаях LDA также может быть реализован как сервер, принимающий от MTA почту по протоколу, аналогичному SMTP . Этот протокол описан в RFC 2033, он называется LMTP (Local Mail Transfer Protocol).

Агентом доставки называется программа, производящая обработку поступившей почты. В основном эта обработка заключается в помещении сообщений в почтовые ящики адресатов, то есть в добавлении сообщений к соответствующим файлам или в размещении их в специальных каталогах пользователей или в базах данных. Пользователь сможет получить сохраненные сообщения, соединившись с хранилищем сообщений по протоколу POP3 или IMAP.

Другой вид обработки сообщений – передача их каким-либо программам для дальнейшей обработки.

Для выполнения этих функций LDA должен при необходимости раскрывать почтовые псевдонимы и списки рассылки.

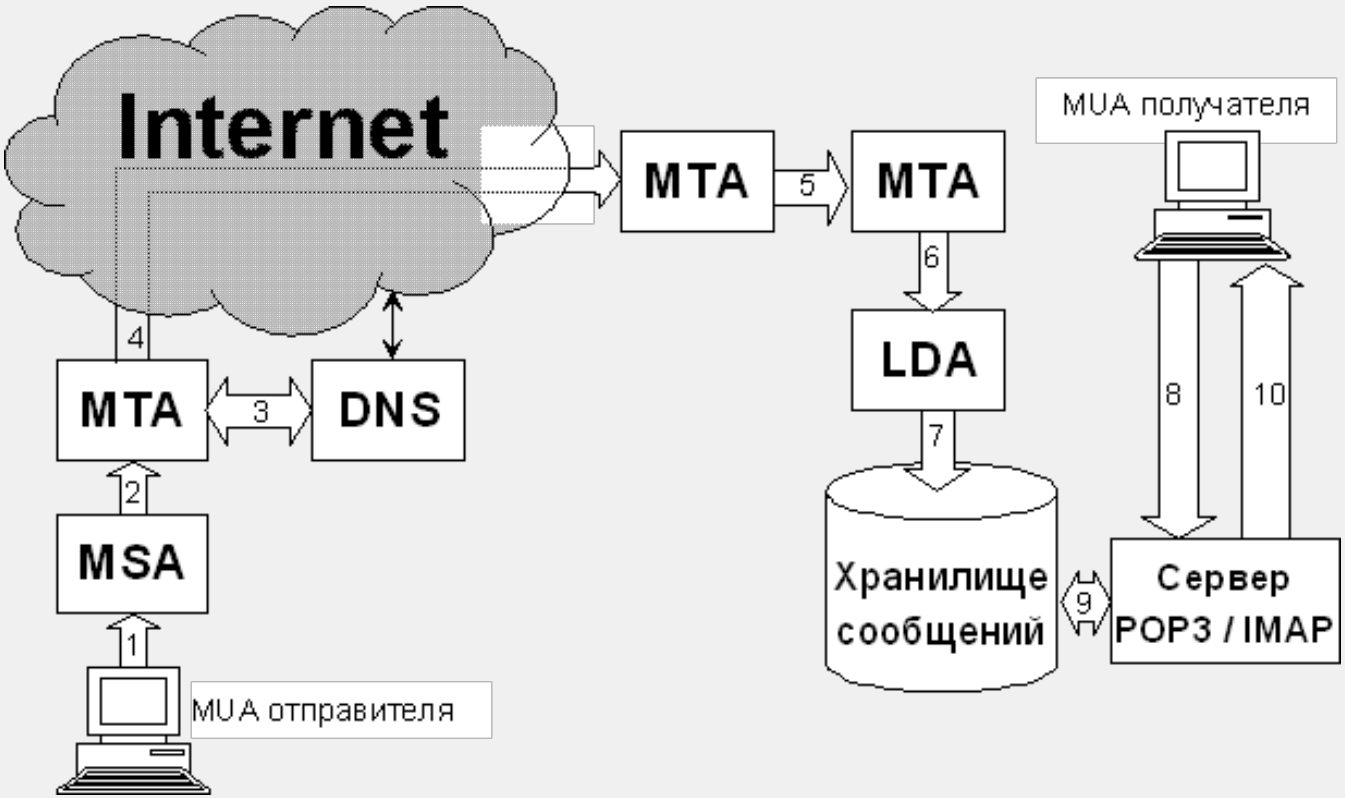
Хранилище сообщений

Электронные сообщения обычно не доставляются автоматически на машину пользователя, а помещаются в хранилище сообщений, откуда пользователь может их забрать в удобное для него время. Каждому пользователю выделяется ограниченный или неограниченный объем дискового пространства, физически реализованный в виде файла специального формата, каталога специальной структуры или набора записей в базе данных. Элемент хранилища сообщений, содержащий электронные сообщения, называется почтовым ящиком.

Доступ пользователей к сообщениям, находящимся в хранилище, обычно осуществляется по протоколам POP3 или IMAP. В качестве клиента выступает MUA пользователя, сервер имеет непосредственный доступ к хранилищу сообщений. Он ожидает подключений пользовательских агентов и, после обязательной аутентификации, определяет права доступа, установленные для данного пользователя. Пользователь должен иметь доступ не менее чем к одному почтовому ящику.

Какие именно манипуляции пользователь может проделывать со своими почтовыми ящиками и с содержащимися в них сообщениями, зависит от используемого программного обеспечения. При минимальной реализации пользователь получает доступ к одному почтовому ящику, сообщения в который помещаются LDA. Пользователь может получать и удалять отдельные сообщения. Такой вид доступа, в большом числе случаев достаточный, реализуется при использовании протокола POP3. Другой популярный протокол доступа к электронным почтовым ящикам – IMAP, предоставляет более широкие возможности.

Доставка почтового сообщения



Доставка почтового сообщения

1. Сообщение, сформированное MUA отправителя, по протоколу SMTP посылается MSA. MSA проверяет, имеет ли данный MUA или пользователь право посылать почту из этой почтовой системы. В случае положительного результата, сообщение принимается для дальнейшей доставки.
2. MSA проверяет заголовок сообщения и, при необходимости, исправляет его. Готовое к отправке сообщение по протоколу SMTP отправляется на MTA исходящей почты.
3. MTA исходящей почты анализирует адрес получателя. Если сообщение предназначено для получателя домена, обслуживаемого данной почтовой системой, то оно доставляется получателю (пункты 6–10), в противном случае MTA запрашивает информацию о почтовом домене, указанном в адресе получателя, сервер DNS. Получив запрашиваемые данные, сервер DNS сообщает MTA, какие узлы принимают почту для данного домена, их адреса IP и приоритеты.
4. MTA отправителя пытается установить соединение по протоколу с принимающими почту узлами в соответствии с приоритетами, указанными в записях MX, полученных от сервера DNS. Если соединение ни с одним узлом не удается установить, сообщение помещается в очередь, и через некоторое время попытки установить соединение повторяются. Если соединение установлено, то принимающий MTA, удостоверившись, что сообщение предназначено для пользователя его домена, и что почтовый ящик с указанным адресом действительно существует, принимает сообщение.
5. В принимающей почтовой системе сообщение может пройти через несколько промежуточных MTA, выполняющих различные виды обработки входящей почты: проверку на вирусы, фильтрацию спама, перенаправление к нужному хранилищу сообщений и пр. Внутри принимающей системы может использоваться как SMTP, так и LMTP.
6. Последний MTA, используя межпроцессное взаимодействие или протокол LMTP, передает сообщение LDA для локальной доставки.
7. LDA помещает сообщение в почтовый ящик адресата.
8. Получатель обращается к серверу POP3 или IMAP, чтобы проверить поступившую почту.
9. Сервер забирает сообщение из почтового ящика и посылает его пользовательскому агенту получателя.

Протокол SMTP

Simple Mail Transfer Protocol (SMTP)

Простой протокол передачи почты. Ообычно используется на участке от MUA отправителя до ближайшего к получателю MTA.
Протокол разрабатывался в начале восьмидесятых годов прошлого века. Окончательная версия была закреплена в RFC 821 1 августа 1982 года. Все годы, прошедшие с того времени, протокол SMTP оставался одним из наиболее часто используемых протоколов семейства TCP/IP. В 2001 был выпущена вторая версия описания протокола — RFC 2821, а в 2008 — третья версия (RFC 5321), которая и используется по настоящее время.

За это время принципиально изменились многие требования, касающиеся достоверности и защищенности передаваемых сообщений, значительно увеличились средний размер сообщений, и их количество, разнообразнее стала передаваемая информация: это уже не только текстовые сообщения на английском языке – сейчас электронные письма пишутся на многих языках и могут содержать вложения самых разных типов.
Однако протокол SMTP получил за время своего существования такое широкое распространение, что просто заменить его другим протоколом уже не представляется возможным. Вместо этого для него разрабатываются различные расширения (extensions), дополняющие возможности базового протокола. Дополненный расширениями протокол SMTP часто называют ESMTP (Extended SMTP).
Сам протокол изменился незначительно. На смену команде HELO , использовавшейся для начала диалога, пришла команда EHLO , позволяющая работать с расширениями ESMTP . Команды, применяемые для настройки почтовых систем и для получения справочной информации о пользователях, теперь используются значительно осторожнее, чем в восьмидесятые годы. Эти команды создают удобства не только для сетевых администраторов, но и для злоумышленников. Поэтому такие команды обычно используют только на этапе настройки почтовой системы. В работающей системе их, как правило, отключают.

Протокол SMTP. Команды SMTP

SMTP обычно использует TCP (25 порт). хотя может работать и с другими протоколами транспортного уровня.
Почта по протоколу SMTP посылается от клиента к серверу. Клиент запрашивает соединение с сервером. После успешного установления соединения сервер сообщает клиенту свое доменное имя. Он также может сообщить тип и версию установленного программного обеспечения. Однако, из соображений безопасности, чтобы не дать потенциальному взломщику воспользоваться известными ошибками данной версии сервера SMTP , передача этой информации часто блокируется системными администраторами.
Ответ сервера, свидетельствующий о готовности к приему команд клиента, служит сигналом к началу диалога, в котором клиент последовательно посылает серверу команды и ожидает ответы, либо подтверждающие исполнение команд, либо сообщающих о невозможности исполнения, либо содержащих информацию, запрошенную клиентом.

Команды SMTP

Каждая команда SMTP начинается с ключевого слова – названия команды. За ним могут следовать параметры, отделенные пробелом. Названия команд и, за редким исключением, параметры протокола, не зависят от регистра. В некоторых элементах расширений строчные и прописные символы могут различаться. Левая часть почтового адреса, до символа «@», может быть регистрозависимой. В командах допускается использование только семибитной кодировки ASCII — цифры, латинские буквы, и знаки препинания. Если информация передается восьмибитными блоками (октетами), старший бит должен быть равен нулю. Корректная интерпретация символов, старший, восьмой бит которых равен единице, например, русских букв, не гарантируется, использовать такие символы не следует.
Конец строк в протоколе SMTP обозначается последовательностью символов "возврат каретки"(`0x0D`) и "перевод строки"(`0x0A`). Эта последовательность обозначается CRLF. Сервер начинает выполнение команды только получив от клиента строку, завершающуюся последовательностью CRLF.
Сервера SMTP должны принимать командные строки длиной до 512 символов. Это значение может быть увеличено по желанию разработчиков. Для серверов, поддерживающих расширения ESMTP, требующие дополнительных параметров, максимально допустимая длина командной строки увеличивается. Соответствующие требования приведены в RFC, описывающих эти расширения. Если не используется расширение, позволяющее серверу принимать несколько команд подряд, клиент передает серверу следующую команду только после получения ответа на предыдущую.

Ответы сервера SMTP

На каждую команду клиента сервер посылает ответ, состоящий из числового кода и отделенной от него пробелом текстовой строки. В большинстве случаев для правильной интерпретации ответа клиенту достаточно числового кода. Текстовая строка нужна для интерпретации ответа человеком. Исключение составляет ответ на команду EHLO, содержащий список расширений ESMTP, поддерживаемых сервером, а так же ответы на некоторые команды ESMTP. Согласно RFC 2821, код ответа состоит из трех цифр. Первая цифра кода может принимать следующие значения:

1 Предварительный положительный результат. Команда принята, но для ее выполнения сервер ожидает реакции клиента на посылаемую в этом ответе информацию. Клиент должен послать следующую команду для продолжения работы. В базовом протоколе SMTP не предусмотрено команд, требующих ответов такого типа.

2 Команда выполнена успешно.

3 Промежуточный положительный результат. Команда принята, но сервер ожидает от клиента дополнительные данные для завершения операции. Дополнительными данными может, например, быть текст сообщения в команде DATA.

4 Исполнение команды временно невозможно. Команда не может быть выполнена, но проблема может быть устранена. Клиенту следует попытаться повторить попытку через некоторое время.

5 Исполнение команды невозможно.

Вторая цифра может принимать следующие значения:

0 Синтаксическая ошибка, неправильное или недопустимое использование команды.

1 Ответ содержит запрошенную информацию.

2 Ответ о состоянии канала передачи.

5 Ответ информирует о состоянии принимающей почтовой системы.

Если ответ состоит из нескольких строк, то каждая из них начинается числовым кодом, который отделяется от сопровождающего текста не пробелом, а символом «дефис» (-). В последней строке цифровой код отделяется от текста пробелом. Каждая строка ответа заканчивается последовательностью CRLF.

Пример диалога SMTP

Команды клиента помечены буквой C, а ответы сервера – буквой S.

S	`220 foo.com Service Ready`	Сервер представляется как `foo.com` и сообщает о готовности к приему команд
C	`EHLO bar.com`	Клиент представляется как `bar.com`
S	`250-foo.com greets bar.com`	
S	`250-8 BITMIME`	Сервер сообщает о поддерживаемых расширениях ESMTP
S	`250-SIZE`	
S	`250-DSN`	
S	`250 HELP`	
C	`MAIL FROM:<Smith@bar.com>`	Адрес отправителя: `Smith@bar.com`
S	`250 OK`	
C	`RCPT TO:<Jones@foo.com>`	Адрес первого получателя: `Jones@foo.com`
S	`250 OK`	
C	`RCPT TO:<Green@foo.com>`	Адрес второго получателя: `Green@foo.com`
S	`550 No such user here`	Ошибка: ящик не существует
C	`RCPT TO:<Brown@foo.com>`	Адрес третьего получателя: `Brown@foo.com`
S	`250 OK`	
C	`DATA`	Адреса переданы, клиент готов передавать сообщение
S	`354 Start mail input; end with <CRLF>.<CRLF>`	Сервер готов к приему сообщения
C	Клиент передает сообщение	Сообщение заканчивается строкой, состоящей из одной точки
C	`.`	
S	`250 OK`	Сообщение принято
C	`QUIT`	Клиент завершает связь
S	`221 foo.com Service closing transmission channel`	Сервер подтверждает завершение связи

Расширения ESMTP

Механизм расширений ESMTP позволяет дополнять протокол SMTP новыми функциональными возможностями, не предусмотренными в RFC 2821. Расширения могут добавлять к протоколу SMTP новые функции или модифицировать существующие. При этом должна сохраняться обратная совместимость: функции базового протокола SMTP должны выполняться независимо от установленных расширений.

Многие расширения ESMTP описаны в документах RFC. Разработчики программного обеспечения также могут использовать в своих продуктах нестандартизованные расширения. Естественно, работать с ними могут только программы того же производителя. Чтобы не допустить ситуации, при которой новое стандартное расширение получит название, которое уже было использовано каким-либо производителем, названия таких расширений должны начинаться с буквы `X`. Название стандартного расширения с буквы `X` начинаться не может.

Расширения ESMTP могут добавлять новые команды, не предусмотренные базовым протоколом SMTP, а также вводить дополнительные параметры команд `MAIL` и `RCPT`. Формат дополнительных параметров:

```
Название_параметра=аргумент
```

Клиент узнает, какие именно расширения поддерживаются сервером, из ответа на команду `EHLO`. Каждая строка ответа может содержать ключевое слово, соответствующее названию поддерживаемого сервером расширения ESMTP, и, если необходимо, параметры этого расширения.

Протокол LMTP

Протокол LMTP (Local Mail Transfer Protocol)

Протокол LMTP применяется в основном для связи MTA с LDA, а также может использоваться для взаимодействия с MTA, не помещающими сообщения в исходящую очередь, и имеющими возможность немедленно ответить, возможна доставка или нет. Протокол LMTP определен в RFC 2033. LMTP работает аналогично SMTP, использует расширения ESMTP, но область его применения отличается от области применения SMTP, и он не должен использовать порт TCP 25.

Отличия LMTP от SMTP

- команды HELO и EHLO заменяются командой LHLO, идентичной по синтаксису и действию команде EHLO;
- команды DATA и, если используется расширение ESMTP CHUNKING, BDAT LAST возвращают не один ответ после окончания приема сообщения, а столько, сколько было успешно выполненных команд RCPT. Сервер передает клиенту результат доставки сообщения каждому получателю.

Пример диалога LMTP

Команды клиента помечены буквой C, а ответы сервера – буквой S.

```
S   220 foo.edu LMTP server ready
C   LHLO foo.edu
S   250-foo.edu
S   250-PIPELINING
S   250 SIZE
C   MAIL FROM:<chris@bar.com>
S   250 OK
C   RCPT TO:<pat@foo.edu>
S   250 OK
C   RCPT TO:<green@foo.edu>
S   250 OK
C   DATA
S   354 Start mail input; end
    with <CRLF>.<CRLF>
C   Передается сообщение
C   .
S   250 OK                          Сообщение успешно доставлено первому адресату: pat@foo.edu
S   452 <green@foo.edu> is          Сообщение не доставлено получателю green@foo.edu
    temporarily over quota
C   QUIT
S   221 foo.edu closing
    connection
```

Протокол POP3

Post Office Protocol — Version 3 (POP3)

Протокол POP3 предназначен для получения сообщений, находящихся в почтовом ящике пользователя на удаленном сервере электронной почты. Как правило, не целесообразно устанавливать серверы SMTP на рабочие станции, предназначенные для чтения писем. Сервер SMTP должен быть доступен постоянно, а рабочие станции обычно включают только на время работы пользователя, соединение с сервером они нередко устанавливают по коммутируемым линиям только для того, чтобы забрать накопившуюся почту.

По протоколу SMTP почта доставляются только в хранилище сообщений, откуда пользователь может ее забрать в удобное для него время.

Таким образом, в качестве клиента POP3 выступает MUA пользователя, а сервер должен иметь доступ к хранилищу сообщений. Информация по протоколу POP3 передается от сервера к клиенту.

Протокол POP был разработан в 1984 году, в 1985 году появилась вторая его версия, в 1988 году – третья, которая с существенными модификациями, сделанными в 1991, 1993, 1994 и 1996 годах, используется по сей день. На момент написания этого пособия, последняя модификация протокола POP3 описана в RFC 1939 и является стандартом Интернет STD 53.

POP3 прост в реализации и предоставляет минимальные необходимые возможности для работы с почтовым ящиком. Вопреки распространенному мнению, третья версия протокола POP дает возможность работать не только с ящиком в целом, но и с отдельными сообщениями, находящимися в нем, позволяя просматривать информацию о письмах, получать и удалять их по отдельности. К сожалению, не все существующие MUA используют эти возможности протокола POP3. Пользователь не всегда хочет скачивать с сервера все содержимое почтового ящика, и часто предпочел бы получать только некоторые сообщения, а другие сообщения, возможно, удалил бы, не получая. Все это можно сделать, используя протокол POP3. Более широкие возможности предоставляет протокол IMAP 4 , но в большом числе случаев возможностей протокола POP3 оказывается вполне достаточно.

Этапы (состояния) сеанса протокола POP3

Сеанс протокола POP3 делится на три этапа (состояния).

Сервер ожидает соединения по порту TCP 110.

После установления соединения сервер посылает клиенту строку приветствия, свидетельствующую о готовности к диалогу, и сеанс переходит в состояние авторизации (AUTHORIZATION State). На этом этапе выясняется, доступ к какому именно почтовому ящику запрашивает клиент и имеет ли он соответствующие права. Успешное прохождение авторизации необходимо для продолжения работы.

Если авторизация проходит успешно, то сеанс переходит в состояние транзакции (TRANSACTION State). На этом этапе клиент может проделывать все необходимые манипуляции с почтовым ящиком: он может просмотреть информацию о состоянии ящика и отдельных сообщений, получить выбранные сообщения и пометить письма, подлежащие удалению.

По окончании всех операций, клиент сообщает об окончании связи, и сеанс переходит в состояние обновления (UPDATE State). На этом этапе сервер стирает из ящика сообщения, помеченные на предыдущем этапе как подлежащие удалению, и закрывает соединение. Переход в состояние обновления в принципе возможен, только если клиент выходит из состояния транзакции по команде QUIT . Ни при каких других обстоятельствах, например, если сеанс связи прерывается по таймауту или из-за обрыва связи, переход в состояние обновления происходить не должен. То есть, если состояние транзакции прерывается не по команде QUIT, никакие удаления не должны производиться, пометки для удаления должны быть аннулированы. Как показывает практика, это требование выполняется не всегда.

Команды протокола POP3

В ходе сеанса клиент посылает серверу команды, а сервер сообщает о результате выполнения каждой из них. Ответ состоит из индикатора состояния (status indicator) и, если нужно, дополнительной информации, отделенной пробелом. Строка ответа может содержать до 512 символов, включая последовательность CRLF, обозначающую конец строки.
Предусмотрено два индикатора состояния: «`+OK`» – успешное завершение и «`-ERR`» – неуспешное завершение. Если строка ответа не содержит дополнительной информации, то после индикатора состояния сразу должна идти последовательность CRLF. Однако некоторые клиенты ожидают пробела после индикатора состояния. Это противоречит существующим стандартам, но наличие таких клиентов все же следует принимать во внимание (RFC 1957).
Если команда предусматривает многострочный ответ, то индикатор состояния передается только в первой строке, а последняя строка ответа должна состоять из одной точки. Эта строка не является частью ответа, а только обозначает его завершение. Чтобы сделать возможным использование строк, состоящих из одной точки, в ответах сервера, ко всем строкам ответа, начинающимся с точки, добавляется еще одна точка, аналогично тому, как это делается при передаче текста сообщения в команде DATA протокола SMTP. Если на приемном конце в ответе сервера обнаруживается строка, начинающаяся с точки, то, если непосредственно за этой точкой стоит последовательность CRLF, строка интерпретируется как конец ответа, если же за точкой следуют любые другие символы, то ведущая точка удаляется, а строка интерпретируется как часть ответа.
Каждая команда POP3 состоит из ключевого слова и, возможно, из аргументов, разделенных пробелами. Ключевые слова состоят из трех или четырех букв, передаваемых независимо от регистра. Аргументы могут содержать только символы ASCII. Каждый аргумент может состоять не более чем из сорока символов.
Кроме обязательных команд программное обеспечение, реализующее взаимодействие по протоколу POP3, поддерживает дополнительные возможности (capabilities), вводящие новые команды, влияющие на исполнение основных команд, облегчающие взаимодействие клиента и сервера, информирующие об особенностях реализации сервера и хранилища сообщений.
В число дополнительных возможностей входят, например, команды авторизации. Хотя бы один механизм авторизации должен быть реализован, так как доступ к почтовому ящику предоставляется только после аутентификации. Но, поскольку таких механизмов несколько, и их выбор оставляется на усмотрение разработчиков и администраторов, соответствующие команды не входят в число обязательных.

Пример сеанса POP3 — 1

S	`+OK ready <6584.1077893295@myhost.ru>`	В начальном приветствии сервера присутствует уникальный идентификатор, что свидетельствует о поддержке сервером команды APOP
C	`CAPA`	Запрос возможностей сервера
S	`+OK Capability list follows`	
S	`TOP`	Поддерживается команда TOP
S	`USER`	Поддерживается авторизация открытым паролем
S	`LOGIN-DELAY 0`	На сервере может быть установлен минимальный период времени между сеансами одного пользователя, но в настоящее время такого ограничения нет
S	`EXPIRE 0`	Прочитанная почта не хранится на сервере (если эта возможность правильно настроена)
S	`UIDL`	Поддерживается команда UIDL
S	`RESP-CODES`	Поддерживаются расширенные коды ответов
S	`X-LOCALTIME Fri, 27 Feb 2004 15:48:46 +0100`	Нестандартное сообщение, установленное разработчиком сервера
S	`.`	Конец ответа
C	`USER lonk`	Имя пользователя: lonk
S	`+OK Password required for lonk.`	Имя принято, ожидается ввод пароля
C	`PASS my_passwd`	Пароль пользователя lonk : my_passwd
S	`+OK lonk has 3 visible messages in 223385 octets.`	Доступ к почтовому ящику разрешен. Имеется 3 сообщения общим объемом 223385 октетов
C	`CAPA`	Повторный запрос возможностей сервера
S	`+OK Capability list follows`	
S	`TOP`	
S	`USER`	

Пример сеанса POP3 — 2

S	LOGIN-DELAY 0	
S	EXPIRE 0	
S	UIDL	
S	RESP-CODES	
S	X-LOCALTIME Fri, 27 Feb 2004 15:49:04 +0100	
S	IMPLEMENTATION Qpopper-version-4.0.3	Строка IMPLEMENTATION доступна только авторизованным пользователям
S	.	Конец ответа
C	LIST	Запрос списка сообщений
S	+OK 3 visible messages (223385 octets)	3 сообщения, 223385 октетов
S	1 111293	Размер первого сообщения: 111293 октета
S	2 111285	Размер второго сообщения: 111295 октетов
S	3 807	Размер третьего сообщения: 807 октетов
S	.	Конец списка
C	UIDL	Запрос списка идентификаторов сообщений
S	+OK uidl command accepted.	
S	1 44b61790f4bb2	Идентификаторы
S	2 0e9c69b8c1feb	
S	3 e30d593d3af4c	
S	.	Конец списка
C	TOP 3 1	Запрос первой строчки третьего сообщения
S	+OK Message follows	
S	Return-Path: <lonk@pds.sut.ru>	Передается заголовок сообщения. Пустая строка – конец заголовка
S	...	

Пример сеанса POP3 — 3

S	`Status: RO`	
S		
S	`Привет!`	Первая строка сообщения
S	`.`	Конец ответа
C	`DELE 3`	Удалить третье сообщение
S	`+OK Message 3 has been deleted.`	Сообщение удалено (на самом деле только помечено для удаления)
C	`STAT`	Запрос количества сообщений
S	`+ OK 2 222578`	Осталось два сообщения
C	`RETR 3`	Запрос третьего сообщения
S	`-ERR Message 3 has been deleted.`	Невозможно получить сообщение, помеченное для удаления
C	`RSET`	Отмена всех удалений
S	`+OK Maildrop has 3 messages (223385 octets)`	В ящике снова три сообщения
C	`RETR 3`	Запрос третьего сообщения
S	`+OK 807 octets`	
S	`Return-Path: <lonk@pds.sut.ru>`	Передается заголовок сообщения
S	`...`	
S	`Status: RO`	
S		
S	`Привет!`	Передается полный текст сообщения
S	`Это тестовое сообщение.`	
S	`.`	
C	`QUIT`	Конец работы
S	`+OK Pop server at myhost.ru signing off.`	

Протокол IMAP

Область применения протокола IMAP (Internet Message Access Protocol) аналогична области применения протокола POP3: он тоже предназначен для получения почты и используется на участке между MUA получателя и хранилищем сообщений. IMAP предоставляет более широкие возможности работы с почтовыми ящиками, чем POP3: он позволяет работать с несколькими почтовыми ящиками на одном или нескольких серверах IMAP как с файлами и каталогами на собственной машине пользователя. Обычно почтовые ящики сервера IMAP действительно представляют собой файлы в специальном каталоге сервера и его подкаталогах.
Сервер IMAP способен анализировать сообщение: выделять заданные поля заголовка и разбирать структуру тела сообщения.
В отличие от серверов POP3, серверы IMAP не должны блокировать ящик на время сеанса – несколько клиентов могут одновременно работать с одним и тем же ящиком. Множественный доступ к почтовым ящикам связан с рядом проблем, особенно, если информация в ящиках доступна для записи. Различные способы разрешения этих проблем описаны в RFC 2180.
Довольно часто IMAP используется в организациях, где пользователям нужно предоставить возможность совместно работать с одними и теми же почтовыми ящиками. Он удобен для работы с новостями USENET. Также протокол можно использовать для работы с личными каталогами и файлами пользователя, расположенными на сервере. Впрочем, для этой цели целесообразнее использовать протоколы, специально предназначенные для работы с каталогами на файловом сервере.

Хотя программное обеспечение, реализующее протокол IMAP, постоянно совершенствуется, IMAP менее защищен, чем POP3. Возможность хранить сообщения на сервере может стать причиной злоупотреблений со стороны пользователей, которые будут переполнять хранилище сообщений ненужной информацией.
Протокол IMAP предполагает в основном работу пользователей с почтовыми ящиками непосредственно на сервере, в отличие от протокола POP3, который ориентирован на то, что клиент забирает пришедшую почту и разбирает ее уже на своей машине (RFC 1733). Это делает IMAP неудобным для пользователей, подключающихся к сети кратковременно, только для того, чтобы получить или отослать почту. Во всяком случае, многие преимущества IMAP таким пользователям недоступны. При работе по протоколу IMAP клиенту желательно иметь доступ к сети все время, пока он работает с почтой.

Протокол IMAP

Протокол IMAP позволяет пользователю работать с множеством почтовых ящиков, расположенных, возможно, на разных серверах.
Допускается иерархическое расположение почтовых ящиков в каталогах и их подкаталогах, причем имена каталогов и почтовых ящиков сами по себе не различаются. Почтовый ящик может быть только конечным элементом иерархической структуры, он не может содержать никаких нижестоящих элементов. Каталог может содержать подкаталоги и почтовые ящики, но он не содержит сообщений и не может быть выбран командой `SELECT`.
Символ, используемый в качестве иерархического разделителя, может различаться в зависимости от используемого на сервере программного обеспечения:

- косая черта («/»), если сервер работает под управлением операционной системы, совместимой с UNIX;
- обратная косая черта («\») для операционной системы Windows:
- точка («.») для имен групп новостей USENET.

Чтобы использовать и различать разные пространства имен на одном сервере IMAP, имена, принадлежащие каждому из используемых пространств, должны начинаться с некоторого префикса, обычно начинающегося символом «#». Естественно, запросы, в которых путь к ящику начинается с одного префикса, будут давать отличные результаты от таких же запросов, начинающихся с другого префикса. Используемое по умолчанию пространство имен может префикса не иметь.
Клиент может выяснить, какие именно пространства имен для почтовых ящиков каких типов поддерживаются данным сервером IMAP, если сервер поддерживает расширение NAMESPACE. Префикс и иерархический разделитель конкретного имени почтового ящика или каталога можно выяснить при помощи команды `LIST`.

Большие возможности протокола IMAP создают большие сложности при разработке, настройке и эксплуатации серверов и клиентов. Некоторые рекомендации по этим вопросам даны в RFC 2683. В общем случае можно посоветовать использовать протокол IMAP только в том случае, если возможности протокола POP3 не достаточны для работы пользователей с их почтовыми ящиками.

Сеанс протокола IMAP

Сервер IMAP ожидает соединения от клиентов на порту TCP 143. После установления соединения сервер посылает свое приветствие клиенту, и начинается диалог, в котором клиент посылает серверу команды, а сервер сообщает о результатах их выполнения или присылает затребованную клиентом информацию. Как и сеанс POP3, сеанс IMAP делится на несколько состояний (states). Допустимый набор команд зависит от текущего состояния сеанса. Сеанс может находиться в одном из следующих состояний:

1. Неаутентифицированное состояние (Not Authenticated State): клиент должен пройти процедуру аутентификации прежде, чем сможет выполнять большинство команд.
2. Аутентифицированное состояние (Authenticated State): клиент аутентифицирован и должен выбрать почтовый ящик, прежде чем сможет работать с отдельными сообщениями.
3. Выбранное состояние (Selected State): почтовый ящик выбран;
4. Состояние выхода (Logout State): сеанс завершается.

Схема переходов между состояниями сеанса IMAP представлена на рисунке.
Переходы, обозначенные цифрами:

1. Соединение без предварительной аутентификации.
2. Соединение с предварительной аутентификацией.
3. Отвергнутое соединение.
4. Успешная аутентификация.
5. Успешное выполнение команды `SELECT` или `EXAMINE`.
6. Команда `CLOSE` или неудачное завершение команды `SELECT` или `EXAMINE`.
7. Команда `LOGOUT` или потеря связи.

Команды клиента и ответы сервера IMAP

Команда клиента состоит из идентификатора (ярлыка) – короткой строкой, состоящей из букв и цифр, не повторяющейся в других командах в течение всего сеанса. За ярлыком следует сама команда и ее аргументы. Регистр символов в названиях команд, как и в большинстве аргументов, как правило, не имеет значения.
Кроме стандартных команд, которые обязательно должны поддерживаться, имеются также дополнительные команды, описанные в стандартах и поддерживаемые серверами IMAP как элементы расширений. Разработчики также могут добавлять в своих реализациях новые команды. Названия таких нестандартизированных команд должны начинаться с буквы «`X`». Имена стандартных команд с буквы «`X`» начинаться не могут.
Все ответы сервера начинаются с метки, после которой следует отделенный пробелом текст.
В ответах сервера, сообщающих об исполнении команд, в качестве метки используется ярлык соответствующей команды. Это помеченные (tagged) ответы. За ним следует одно из ключевых слов:

- `OK` (успешное выполнение);
- `NO` (невыполнение);
- `BAD` (ошибка в команде).

Источники

- Материалы с сайта `https://wikipedia.org/`
- Telecommunication technologies — телекоммуникационные технологии / Ю. А. Семенов. URL: `http://book.itep.ru/`
- RFC 5321. Simple Mail Transfer Protocol.
- RFC 1939. Post Office Protocol - Version 3.
- RFC 3501. Internet Message Access Protocol - Version 4rev1.
- Электронный курс «Структура и протоколы электронной почты в INTERNET».