

# Лабораторная работа № 7

## Прием почтовых сообщений по протокол POP3

### 7.1 Цель работы

Ознакомиться с технологией работы POP3 клиента и реализация его с базовым набором команд.

### 7.2 Задание на лабораторную работу

Разработать приложение, позволяющее принимать электронные письма с вложенными файлами, используя протокол POP3.

### 7.3 Методические указания

*Post Office Protocol (POP)* - протокол доставки почты пользователю из почтового ящика *почтового сервера POP*. Многие концепции, принципы и понятия протокола POP выглядят и функционируют подобно SMTP. Команды POP практически идентичны командам SMTP, отличаясь в некоторых деталях. POP3 разработан с учетом специфики доставки почты на персональные компьютеры и имеет соответствующие операции для этого.

#### 7.3.1 Назначение протокола POP3

Ранее почтовые сообщения большинства сетей доставлялись непосредственно от одного компьютера к другому. И если пользователь часто менял рабочие компьютеры или один компьютер принадлежал нескольким пользователям, существовали определенные проблемы. В наши дни общепринята доставка сообщения не на компьютеры пользователя, а в специальные почтовые ящики *почтового сервера* организации, который круглосуточно работает (включен).

В протоколе POP3 оговорены три стадии процесса получения почты: авторизация, транзакция и обновление. После того как сервер и клиент POP3 установили соединение, начинается стадия авторизации. На стадии авторизации клиент идентифицирует себя для сервера. Если авторизация прошла успешно, сервер открывает почтовый ящик клиента и начинается стадия транзакции. В ней клиент либо запрашивает у сервера информацию (например, список почтовых сообщений), либо просит его совершить определенное действие (например, выдать почтовое сообщение). Наконец, на стадии обновления сеанс связи заканчивается. В табл.3 перечислены команды протокола POP3, обязательные для работающей в Интернет реализации минимальной конфигурации.

### 7.3.2 Команды протокола POP3

Таблица 7.1

#### Команды протокола POP версии 3

Команда	Описание
<b>USER</b> <i>имя_пользователя</i>	Идентифицирует пользователя с указанным именем
<b>PASS</b> <i>пароль</i>	Указывает пароль
<b>QUIT</b>	Закрывает TCP-соединение
<b>STAT</b>	Сервер возвращает количество сообщений в почтовом ящике плюс размер почтового ящика после метки "+ОК" выводит два числа: число сообщений и их общий объем в байтах.
<b>LIST</b> <i>n</i>	Сервер возвращает идентификаторы сообщений вместе с размерами сообщений (параметром команды может быть идентификатор сообщения) если <i>n</i> указано, то после метки "+ОК" выводит размер сообщения номер <i>n</i> в байтах. Иначе выводит список из двух колонок: номер сообщения, пробел, размер сообщения в байтах; вывод списка заканчивается строкой, содержащей только символ "." ("точка").
<b>RETR</b> <i>n</i>	Извлекает сообщение из почтового ящика (требуется указывать аргумент-идентификатор сообщения) выводит сообщение номер <i>n</i> . Вывод заканчивается строкой, содержащей только символ "." ("точка").
<b>DELE</b> <i>n</i>	Отмечает сообщение для удаления (требуется указывать аргумент - идентификатор сообщения) при этом нумерация сообщений не изменяется, а все удаленные в данном сеансе сообщения могут быть восстановлены командой REST.

Продолжение таблицы 7.1

Команда	Описание
<b>NOOP</b>	Сервер возвращает положительный ответ, но не совершает никаких действий
<b>LAST</b>	после метки "+OK" выводит номер последнего по времени сообщения, для которого была выполнена команда RETR; эта информация сохраняется между сеансами, что позволяет не запрашивать дважды уже полученные пользователем, но не удаленные с сервера сообщения.
<b>TOP <i>n m</i></b>	выводит заголовок и <i>m</i> первых строк сообщения номер <i>n</i> . Вывод заканчивается строкой, содержащей только символ "." ("точка").
<b>RSET</b>	Отменяет удаление сообщения, отмеченного ранее командой DELE

В протоколе POP3 определено несколько команд, но на них дается только два ответа: +OK (позитивный, аналогичен сообщению-подтверждению ACK) и -ERR (негативный, аналогичен сообщению "не подтверждено" NAK). Оба ответа подтверждают, что обращение к серверу произошло и что он вообще отвечает на команды. Как правило, за каждым ответом следует его содержательное словесное описание. В RFC 1225 есть образцы нескольких типичных сеансов POP3. Сейчас мы рассмотрим несколько из них, что даст возможность уловить последовательность команд в обмене между сервером и клиентом.

### 7.3.3 Пример POP3 транзакции

После того как программа установила TCP-соединение с портом протокола POP3 (официальный номер 110), необходимо послать команду USER с именем пользователя в качестве параметра. Если ответ сервера будет +OK, нужно послать команду **PASS** с паролем этого пользователя:

```
USER user22
+OK Password required for user user22
PASS ****
+OK user22@mail.ru maildrop has 684 messages (108935168 octets)
```

После того как стадия авторизации окончена, обмен переходит на стадию транзакции. В следующих примерах демонстрируется возможный обмен сообщениями на этой стадии.

Команда **STAT** возвращает количество сообщений и количество байтов в сообщениях:

```
CLIENT: STAT  
SERVER: +OK 2 320
```

Команда **LIST** (без параметра) возвращает список сообщений в почтовом ящике и их размеры:

```
CLIENT: LIST  
SERVER: +OK 2 messages (320 octets)  
SERVER: 1 120  
SERVER: 2 200  
SERVER: . ...
```

Команда **LIST** с параметром возвращает информацию о заданном сообщении:

```
CLIENT: LIST 2  
SERVER: +OK 2 200 ...  
CLIENT: LIST 3  
SERVER: -ERR no such message, only 2 messages in maildrop
```

Команда **TOP** возвращает заголовок, пустую строку и первые десять строк тела сообщения:

```
CLIENT: TOP 10  
SERVER: +OK
```

*SERVER: <the POP3 server sends the headers of the message, a blank line, and the first 10 lines of the message body> (сервер POP высылает заголовки сообщений, пустую строку и первые десять строк тела сообщения)*

```
SERVER: . ... CLIENT: TOP 100 SERVER: -ERR no such message
```

Команда **NOOP** не возвращает никакой полезной информации, за исключением позитивного ответа сервера. Однако позитивный ответ означает, что сервер находится в соединении с клиентом и ждет запросов:

```
CLIENT: NOOP  
SERVER: +OK
```

Следующие примеры показывают, как сервер POP3 выполняет действия. Например, команда **RETR** извлекает сообщение с указанным номером и помещает его в буфер местного UA:

*CLIENT: RETR 1*

*SERVER: +OK 120 octets*

*SERVER: <the POPS server sends the entire message here> (POP3-сервер высылает сообщение целиком)*

*SERVER: . . . . .*

Команда **DELE** отмечает сообщение, которое нужно удалить:

*CLIENT: DELE 1*

*SERVER: +OK message 1 deleted ... (сообщение 1 удалено)*

*CLIENT: DELE 2*

*SERVER: -ERR message 2 already deleted (сообщение 2 уже удалено)*

Команда **RSET** снимает метки удаления со всех отмеченных ранее сообщений:

*CLIENT:RSET*

*SERVER: +OK maildrop has 2 messages (320 octets)  
(в почтовом ящике 2 сообщения (320 байтов) )*

Как и следовало ожидать, команда **QUIT** закрывает соединение с сервером:

*CLIENT: QUIT*

*SERVER: +OK dewey POP3 server signing off*

*CLIENT: QUIT*

*SERVER: +OK dewey POP3 server signing off (maildrop empty)*

*CLIENT: QUIT*

*SERVER: +OK dewey POP3 server signing off (2 messages left)*

Обратите внимание на то, что отмеченные для удаления сообщения на самом деле не удаляются до тех пор, пока не выдана команда **QUIT** и не началась стадия обновления. В любой момент в течение сеанса клиент имеет возможность выдать команду **RSET**, и все отмеченные для удаления сообщения будут восстановлены.

#### **7.4 Контрольные вопросы и задания**

1. Объяснить назначение протокола POP3.
2. Объяснить назначение команд **NOOP**, **RETR**, дополнить свою программу этими функциями.
3. Объяснить назначение команд **DELE**, **RSET**, дополнить свою программу этими функциями.
4. Объяснить назначение команд **LIST**, **STAT**, дополнить свою программу этими функциями.