

Клиент протокола POP-3

Задание

Разработать приложение для операционных систем семейства Windows или Linux, обеспечивающее функции клиента протокола POP-3.

Приложение должно реализовывать следующие функции:

- 1) Подключение к указанному серверу по IP-адресу или доменному имени
- 2) Получение состояния ящика(количество новых писем, их суммарная длина)
- 3) Получение списка заголовков всех новых писем сервера без предварительной загрузки
- 4) Загрузка всех новых или конкретных выбранных писем
- 5) Пометка конкретных писем для последующего удаления
- 6) Удаление помеченных писем
- 7) Выход из приложения без удаления помеченных писем 8) Подробное протоколирование соединения сервера с клиентом

Поддерживаемые команды

Разработанное приложение должно реализовывать следующие команды протокола POP-3:

- USER – передача серверу идентификационной информации пользователя
- PASS – передача серверу пароля пользователя
- LIST – получение списка сообщения почтового ящика
- RETR – получение сообщения
- DELE – пометка сообщения на удаление
- TOP – получение первых нескольких строк сообщения
- UIDL – получение уникального идентификатора сообщения
- RSET – сброс всех пометок на удаление сообщений
- QUIT – удаление всех помеченных сообщений и завершение сеанса

Настройки приложения

Разработанное приложение должно предоставлять пользователю настройку следующих параметров:

- 1) IP-адрес или доменное имя почтового сервера
- 2) Номер порта сервера(по умолчанию - 995)
- 3) Имя пользователя
- 4) Пароль пользователя

Архитектура приложения

В приложении имеется класс POP3Client, который взаимодействует с сервером и реализует команды протокола POP3. В этом классе реализованы следующие методы:

- send() - посылает команды серверу
- recv() - получает ответ от сервера
- recv2() - получает ответ от сервера для команд retr и top
- command() - отправляет команды серверу и получает ответ, проверяет что ответ "ОК"
- connect_to_serv() - подключается к серверу, метод аутентификации
- stat(), list(), top(), uidl(), retr(), retr(), dele(), rset(), quit() - реализуют одноименные команды
- par_list(), par_top(), par_uidl(), par_retr(), par_dele() - отвечают за ввод недостающих параметров для соответствующих команд.

Описание работы протокола

Сервер прослушивает TCP соединение на порту 995. Когда клиент желает воспользоваться сервисом POP3, он должен установить соединение с сервером. Клиент и POP3 сервер обмениваются командами и ответами до тех пор, пока соединение не будет закрыто или прервано.

Все команды заканчиваются парой CRLF. Ключевые слова и аргументы состоят из ASCII символов. Ключевые слова и аргументы разделены одиночным пробелом. Ключевые слова состоят из 3-х или 4-х символов, каждый аргумент может быть длиной до 40 символов. В есть два ответа: положительный (+OK) и отрицательный (-ERR). Определенные ответы могут быть многострочными. После того как все строки ответа посланы, последняя строка будет заканчиваться символом «.»

POP3 сессия состоит из нескольких стадий. После установки TCP соединения, сервер посылает приветствие, и сессия переходит в состояние AUTHORIZATION. На этом этапе клиент должен идентифицировать себя на сервере. После успешной идентификации сессия переходит в состояние TRANSACTION. В этой стадии клиент запрашивает выполнение команд на сервере. Когда клиент посылает команду QUIT, сессия переходит в состояние UPDATE. На этом этапе POP3 сервер освобождает все ресурсы, занятые в стадии TRANSACTION и заканчивает работу. TCP соединение после этого закрывается.

Дизайн протокола

Логирование:

После запуска приложения клиенту требуется ввести имя сервера и порта:

Input name of server

Input port

После чего предлагается ввести логин и пароль:

Input username

Input password

При успешной аутентификации, устанавливается соединение между клиентом и сервером.

Клиент может вводить команды согласно протоколу:

Input command

Если пользователь не указал параметры команды, то ниже ему предлагается их ввести:

Input num

Input number of strings

При вводе команды quit анализируется соединение. Если соединение установлено, оно сбрасывается, и выводится сообщение об удачной операции quit.

Описание команд протокола

Команда Описание

STAT	Используется для просмотра состояния текущего почтового ящика. В ответ POP3- сервер возвращает строку, содержащую количество и общий размер в байтах сообщений, которые клиент может получить с POP3- сервера. Сообщения, помеченные на удаление, не учитываются.
LIST [msg]	Получение списка сообщений почтового ящика. Сообщения, помеченные на удаление не фигурируют в этом списке.
RETR msg	Используется для передачи клиенту запрашиваемого сообщения. Аргумент команды — номер сообщения. Если запрашиваемого сообщения нет, возвращается отрицательный индикатор «-ERR».
DELE msg	Аргумент команды— номер сообщения. Сообщения, помеченные на удаление, реально удаляются только после закрытия транзакции, на стадии UPDATE.
RSET	Для отката транзакции внутри сессии используется команда RSET (без аргументов). Если пользователь случайно Пометил на удаление какие-либо сообщения, он может убрать эти пометки, отправив эту команду.
TOP msg n	По этой команде пользователь может получить «n» первых строк сообщения с номером «msg». POP3- сервер по запросу клиента отправляет заголовок сообщения, затем пустую строку, затем требуемое количество строк сообщения (если количество строк в сообщении меньше указанного в параметре «n», пользователю передается все сообщение).
USER name	Когда POP3 -сессия находится в состоянии аутентификации (AUTHORIZATION), и клиент должен зарегистрировать себя на POP3 - сервере. Это может быть выполнено либо с помощью команд USER и PASS — ввод открытых пользовательского идентификатора и пароля (именно этот способ используется чаще), либо командой APOP — аутентификация цифровой подписью, на базе секретного ключа. Любой POP3 -сервер должен поддерживать хотя бы один из механизмов аутентификации. Аргументом — «name» является строка, идентифицирующая почтовый ящик системы. Этот идентификатор должен быть уникальным в данной почтовой системе POP3 -сервера. Если ответом на эту команду является строка индикатора «+OK», клиент может отправлять команду PASS — ввод пароля или QUIT — завершить сессию. Если ответом является строка «-ERR», клиент может либо повторить команду USER, либо закрыть сессию

Аргументом команды является строка пароля данного почтового ящика. После получения команды PASS, POP3 -сервер, на основании аргументов команд USER и PASS, определяет возможность доступа к заданному почтовому ящику. Если POP3 -сервер ответил «+OK», это означает, что аутентификация клиента прошла успешно и он может работать со своим почтовым ящиком, т. е. сессия переходит в состояние TRANSACTION. Если POP3- сервер ответил «-ERR», то либо был введен неверный пароль, либо не найден указанный почтовый ящик.

К командам состояния AUTHORIZATION может относиться команда закрытия POP3- сессии — QUIT , если она была отправлена в режиме AUTHORIZATION (например, при вводе неправильного пароля или идентификатора пользователя): Эта команда отправляется без аргументов и всегда имеет единственный ответ «+OK».

Среда разработки

PyCharm 4.0.4

Python 2.7.9

Тестирование

Проверим подключение к серверу, исполнение всех команд (с параметрами и без):

USER

PASS

LIST

RETR

DELE

TOP

UIDL

RSET

QUIT

Проверим также ввод неверных команд, исполнение команд с неверными параметрами(несуществующими письмами).

Input name of server

pop.yandex.ru

Input port

995

<< +OK POP Ya! na@11 mled3u1L1W2h

Input username

denispopclient1994@yandex.ru

<< +OK password, please.

Input password:

denis11021994

<< +OK 2 22089

Input command

stat

<< +OK 2 22089

Input command

list

Input num 1

<< +OK 1 10477

Input command

list 2

<< +OK 2 11612

Input command

top

Input number of msg

1

Input number of strings

10

<< +OK 10477 octets.

X-Yandex-FolderName: Vhodyashchie Content-Type: multipart/related;
boundary="====1696383123==" MIME-Version: 1.0 From: =?utf-8?b?
0K/QvdC00LXQutGB?= hello@yandex.ru Subject: =?utf-8?b?
0KHQvtCx0LXRgNC40YLQtSDQstGB0Y4g0L/QvtGH0YLRgyDQsiDRjdGC0L4=?=
=?utf-8?b?0YIg0Y/RidC40Lo=?= Message-Id:
20110815165837.A26162B2802A@yaback1.mail.yandex.net

—====1696383123== Content-Type: text/html; charset="utf-8" MIME-
Version: 1.0 Content-Transfer-Encoding: base64

PGh0bWw+PGhlYWQ+PG1ldGEgaHR0cC1lcXVpdj0iY29udGVudC10eXBlliBjb250ZW
L2h0bWw7IGNoYXJzZXQ9VVRGLTgiPjwvaGVhZD48Ym9keSBiZ2NvbG9yPSIjZmZl
PSIjMDAwMDAwIj48bWV0YSBodHRwLWVxdWl2PSJDb250ZW50LVR5cGUiIGNvbr
aHRtbDsgY2hhcnNldD1VVEYtOCI+Cjx0aXRsZT7QodC+0LHQtdGA0LjRgtC1INC/0Lj
sCDQuNC3INC00YDRg9Cz0LjRhSDQv9C+0YfRgtC+0LLRi9GFINGP0YnQuNC60L7C

.

Input command

top 1 10

<< +OK 10477 octets.

X-Yandex-FolderName: Vhodyashchie Content-Type: multipart/related;
boundary="====1696383123==" MIME-Version: 1.0 From: =?utf-8?b?
0K/QvdC00LXQutGB?= hello@yandex.ru Subject: =?utf-8?b?
0KHQvtCx0LXRgNC40YLQtSDQstGB0Y4g0L/QvtGH0YLRgyDQsiDRjdGC0L4=?=
=?utf-8?b?0YIg0Y/RidC40Lo=?= Message-Id:
20110815165837.A26162B2802A@yaback1.mail.yandex.net

—====1696383123== Content-Type: text/html; charset="utf-8" MIME-
Version: 1.0 Content-Transfer-Encoding: base64

PGh0bWw+PGhlYWQ+PG1ldGEgaHR0cC1lcXVpdj0iY29udGVudC10eXBlliBjb250ZW
L2h0bWw7IGNoYXJzZXQ9VVRGLTgiPjwvaGVhZD48Ym9keSBiZ2NvbG9yPSIjZmZl
PSIjMDAwMDAwIj48bWV0YSBodHRwLWVxdWl2PSJDb250ZW50LVR5cGUiIGNvbr
aHRtbDsgY2hhcnNldD1VVEYtOCI+Cjx0aXRsZT7QodC+0LHQtdGA0LjRgtC1INC/0Lj
sCDQuNC3INC00YDRg9Cz0LjRhSDQv9C+0YfRgtC+0LLRi9GFINGP0YnQuNC60L7C

.

Input command

uidl

Input number of msg

1

<< +OK 1 c115536a552aee89e0406c5f16fd716c

Input command

uidl 2

<< +OK 2 a92a82c5fe08b1b032b69906c2b7b19a

Input command
retr 1

<< +OK 10477 octets.

X-Yandex-FolderName: Vhodyashchie Content-Type: multipart/related;
boundary="=====1696383123==" MIME-Version: 1.0 From: =?utf-8?b?
0K/QvdC00LXQutGB?= hello@yandex.ru Subject: =?utf-8?b?
0KHQvtCx0LXRgNC40YLQtSDQstGB0Y4g0L/QvtGH0YLRgyDQsiDRjdGC0L4=?=
=?utf-8?b?0YIg0Y/RidC40Lo=?= Message-Id:
20110815165837.A26162B2802A@yaback1.mail.yandex.net

—=====1696383123== Content-Type: text/html; charset="utf-8" MIME-
Version: 1.0 Content-Transfer-Encoding: base64

.....
tTfFRWbDtdVXMJyiGHuKalWhUxPNpXrbg40qet1umGFABQAUAFAGD0UAZoAON/
BQAUAFABQAUAFAH/2Q== —=====1696383123==— .

Input command
dele 2

<< +OK 1 10477

Input command
stat

<< +OK 1 10477

Input command
rset

<< +OK 2 22089

Input command
avasd

Wrong command

Input command
list 3

<< -ERR message does not exist or deleted

Input command
quit

<< +OK shutting down.

Также, для данного приложения были написаны unit-тесты. Итоговое тестовое покрытие составляет 56%. Тесты приведены в приложении №2.

Приложение 1. Текст программы

```
__author__ = 'Denis'

import socket, base64, ssl, getpass
import ssl
HAVE_SSL = True

class POP3Client:

    def __init__(self, servers, ports):
        self.sock = ssl.wrap_socket(socket.socket(socket.AF_INET, socket.SOCK_ST
        self.server = servers
        self.port = ports

    def send(self, str):
        print(">> " + str)
        self.sock.send(str)

    def recv(self):
        self.answer = self.sock.recv(1024)
        print("<< " + self.answer)

    def recv2(self):
        self.answer = ""
        while not '.' in self.answer.splitlines():
            new_data = self.sock.recv(1024)
            self.answer += new_data
        print("<< " + self.answer)

    def command(self, cmd, wait_dot=False):
        self.send(cmd+"\n")
        if wait_dot:
            self.recv2()
        else:
            self.recv()
        assert self.answer.startswith("+OK")
        return self.answer

    def connect_to_serv(self):
        self.sock.connect((self.server, self.port))
        self.recv()
        self.command("USER denispopclient1994@yandex.ru")
        self.command("PASS denis11021994")
        """print("Input username")
        username = raw_input()
        self.command("USER "+username)
        print ("Input password")
        password = raw_input()
        self.command("PASS "+password)"""

    def stat(self):
```

```

        """cmd = self.command('STAT')
        parse = cmd.split(' ')
        num = int(parse[1])
        size = int(parse[2])
        return num"""
        return self.command('STAT')

def list(self, which=None):
    return str(self.command('LIST %s' % which))

def top(self, which, howmuch):
    return self.command('TOP %s %s' % (which, howmuch), True)

def uidl(self, which=None):
    return self.command('UIDL %s' % which)

def retr(self, which):
    return self.command('RETR %s' % which, True)

def dele(self, which):
    return self.command('DELE %s' % which)

def rset(self):
    return self.command('RSET')

def quit(self):
    return self.command('QUIT')
    self.sock.close()

def par_list(self, num=None):
    if num == None:
        print("Input num")
        num = raw_input()
    self.list(num)

def par_top(self, msg=None, num=None):
    if msg == None:
        print("Input number of msg")
        msg = raw_input()
    if num == None:
        print("Input number of strings")
        num = raw_input()
    self.top(int(msg), int(num))

def par_uidl(self, msg=None):
    if msg == None:
        print("Input number of msg")
        msg = raw_input()
    self.uidl(msg)

def par_retr(self, msg=None):
    if msg == None:
        print("Input number of msg")
        msg = raw_input()
    self.retr(msg)

```

```

def par_dele(self,msg=None):
    if msg == None:
        print("Input number of msg")
        msg=raw_input()
    self.dele(int(msg))

"""def control(self,msg):
    parse = cmd.split(' ')
    num = int(parse[1])
    size = int(parse[2])
    return num
    if (msg>self.connect_to_serv()):
        print ("Such number of the letter doesn't exist")
        return False
    else:
        return True"""

def main():
    """print("Input name of server")
    server=raw_input()
    print("Input port")
    port=raw_input()
    client = POP3Client(server, int(port))"""
    client = POP3Client('pop.yandex.ru', 995)
    client.connect_to_serv()
    commands = {
        'stat': client.stat,
        'list': client.par_list,
        'top': client.par_top,
        'retr': client.par_retr,
        'uidl': client.par_uidl,
        'dele': client.par_dele,
        'rset': client.rset,
        'quit': client.quit,
    }
    cmd=''
    while cmd != 'quit':
        print("Input command")
        cmd = raw_input()
        parse=cmd.split(' ')
        if parse[0] in commands:
            command = commands[parse[0]]
            command(*parse[1:])
        else:
            print("Wrong command")
        # if cmd == 'STAT':
        #     client.stat()
        # elif parse[0] == 'TOP':
        #     if len(parse)==1:
        #         client.print_top()
        #     elif len(parse)==2:
        #         client.print_top(parse[1])
        #     else:
        #         client.print_top(parse[1],parse[2])

```

```
# elif parse[0] == 'LIST':
#     if len(parse)==1:
#         client.print_list()
#     else:
#         client.print_list(parse[1])
# elif parse[0] == 'UIDL':
#     if len(parse)==1:
#         client.print_uidl()
#     else:
#         client.print_uidl(parse[1])

if __name__ == "__main__":
    main()
```

Приложение 2. Unit-тесты

```
import io
from nose.tools import raises
from main import POP3Client

__author__ = 'Denis'

import socket
import unittest

class testPop3(unittest.TestCase):
    def setUp(self):
        self.login = 'denispopclient1994@yandex.ru'
        self.passwd = 'denis11021994'
        self.serv = 'pop.yandex.ru'
        self.port = '995'
        self.client = POP3Client(self.serv, int(self.port))
        self.client.connect_to_serv()
        self.client.user(self.login)
        self.client.passwd(self.passwd)

    def test_dele(self):
        tmp1 = self.client.stat()
        parse1=tmp1.split(' ')[1]
        self.client.dele(1)
        self.client.quit()
        print('1 dele: %s' % str(parse1))
        self.client = POP3Client(self.serv, int(self.port))
        self.client.connect_to_serv()
        self.client.user(self.login)
        self.client.passwd(self.passwd)
        tmp2 = self.client.stat()
        parse2=tmp2.split(' ')[1]
        print('2 dele: %s' % str(parse2))
        self.assertLess(parse2, parse1)

    def test_rset(self):
        tmp1 = self.client.stat()
        parse1=tmp1.split(' ')[1]
        self.client.dele(1)
        self.client.rset()
        self.client.quit()
        self.client = POP3Client(self.serv, int(self.port))
        self.client.connect_to_serv()
        self.client.user(self.login)
        self.client.passwd(self.passwd)
        tmp2 = self.client.stat()
        parse2=tmp2.split(' ')[1]
        self.assertEqual(parse2, parse1)

    @raises(IOError)
    def test_close(self):
```

```

        self.client = POP3Client(self.serv, int(self.port))
        self.client.connect_to_serv()
        self.client.user(self.login)
        self.client.sock.close()
        self.client.passwd(self.passwd)

@raises(IOError)
def test_close2(self):
    self.client = POP3Client(self.serv, int(self.port))
    self.client.connect_to_serv()
    self.client.user(self.login)
    self.client.passwd(self.passwd)
    self.client.sock.close()
    self.client.retr(1)

def test_topic(self):
    #str = open('letter.txt').read()
    str = self.client.retr(1)
    topics = ['From', 'To', 'Date']
    new = str.partition('\r\n\r\n')[0]
    lines = new.splitlines()
    keys = [x.partition(':')[0].strip() for x in lines]
    if set(topics).issubset(set(keys)):
        print('OK')
    else:
        print('Error')
        raise Exception

if __name__ == '__main__':
    unittest.main()

```