МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ ДНР

ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра ПИ

Факультет КНТ

Лабораторная работа №2

Тема: «Изучение протоколов передачи электронной почты»

Курс: Протоколы компьютерных сетей

Выполнил

ст. гр. ПИ-18А

Данильчук К.М.

Проверил

Чернышова А.В.

Донецк – 2021

ЛИСТИНГ КОДА

SMTP

from PyQt5.QtWidgets import QWidget, QMessageBox, QFileDialog

from view.main\_ui import Ui\_mainForm

from authorization import AuthorizationForm

import smtplib

from email import encoders

from email.header import Header

from email.mime.base import MIMEBase

from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

import os

import mimetypes

def attach\_file(filepath, msg):

filename = os.path.basename(filepath)

ctype, enc = mimetypes.guess\_type(filepath)

maintype, subtype = ctype.split('/', 1)

with open(filepath, 'rb') as fp:

file = MIMEBase(maintype, subtype)

file.set\_payload(fp.read())

encoders.encode\_base64(file)

file.add\_header('Content-Disposition', 'attachment', filename=filename)

msg.attach(file)

class MainForm(QWidget, Ui\_mainForm):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.setupUi(self)

self.initUi()

self.authorization = AuthorizationForm()

self.authorization.logined.connect(self.check\_logIn)

def initUi(self):

self.attach\_button.clicked.connect(self.attach\_button\_clicked)

self.clear\_button.clicked.connect(self.clear\_button\_clicked)

self.logOut\_button.clicked.connect(self.logOut\_button\_clicked)

self.send\_button.clicked.connect(self.send\_button\_clicked)

def send\_button\_clicked(self):

msg = MIMEMultipart()

msg.encoding = 'utf-8'

msg['Subject'] = Header(self.subject\_line.text())

msg['From'] = Header(self.login)

msg['To'] = Header(self.to\_line.text())

msg.attach(

MIMEText(self.message\_text.toPlainText(), 'plain', 'utf-8'))

for i in range(0, self.files\_list.count()): # attach files from list

item = self.files\_list.item(i)

attach\_file(item.text(), msg)

try:

self.connect.sendmail(

self.login, self.to\_line.text().split(' '), msg.as\_string())

except:

QMessageBox.warning(self, 'Ошибка', 'Проверте правильность набора')

def logOut\_button\_clicked(self):

self.connect.quit()

self.close()

self.authorization.show()

def clear\_button\_clicked(self):

self.files\_list.clear()

def attach\_button\_clicked(self):

filepath = QFileDialog.getOpenFileName(self, 'Выберите файл')[0]

self.files\_list.addItem(filepath)

def start(self):

self.authorization.show()

def check\_logIn(self, login, password):

try:

self.connect = smtplib.SMTP('smtp.gmail.com', 587)

self.connect.starttls()

self.connect.login(login, password)

except smtplib.SMTPAuthenticationError:

QMessageBox.warning(self, 'Ошибка авторизации',

'Неверный логин или пароль')

except:

QMessageBox.warning(self, 'Ошибка авторизации',

'Введите корректный логин и пароль')

else:

self.login = login

self.authorization.close()

self.show()

if \_\_name\_\_ == '\_\_main\_\_':

from PyQt5.QtWidgets import \*

import sys

app = QApplication(sys.argv)

w = MainForm()

w.show()

sys.exit(app.exec())

from PyQt5.QtWidgets import QWidget

from PyQt5.QtCore import pyqtSignal

from view.authorization\_ui import Ui\_AuthorizationForm

class AuthorizationForm(QWidget, Ui\_AuthorizationForm):

logined = pyqtSignal(str, str)

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.setupUi(self)

self.logIn\_button.clicked.connect(self.logIn\_button\_click)

def logIn\_button\_click(self):

self.logined.emit(self.login\_line.text(), self.password\_line.text())

# from PyQt5.QtWidgets import \*

# import sys

# app = QApplication(sys.argv)

# w = AuthorizationForm()

# w.show()

# sys.exit(app.exec())

from PyQt5.QtWidgets import QApplication

import sys

from mainForm import MainForm

if \_\_name\_\_ == '\_\_main\_\_':

app = QApplication(sys.argv)

window = MainForm()

window.start()

sys.exit(app.exec())

POP3

import poplib

from getpass import getpass

import mailparser

import sys

if \_\_name\_\_ == '\_\_main\_\_':

login = input('login: ')

password = getpass('password: ')

if not login and not password:

login = 'silurcpppy@gmail.com'

password = 'smtplib1'

server = poplib.POP3\_SSL('pop.gmail.com', 995)

try:

server.user(login)

server.pass\_(password)

except BaseException:

print("Wrong login or password.")

sys.exit(-1)

response, message\_sizes, octets = server.list()

message\_count, \_ = server.stat()

print(f'Message count: {message\_count}')

for msgnum, msgsize in [i.decode().split() for i in message\_sizes]:

resp, lines, octets = server.retr(msgnum)

msgtext = b"\n".join(lines) + b"\n\n"

mail = mailparser.parse\_from\_bytes(msgtext)

print(f"Message: {msgnum}")

print(f"От: {' '.join(mail.from\_[0])}")

print(f'Кому: {mail.delivered\_to}')

print(f'Тема: {mail.subject}')

print(f'Дата: {mail.date.replace(hour=mail.date.hour + 3)}')

print("Сообщение:\n" + ''.join(mail.text\_plain))

print("Файлы:\n" + ' '.join([file['filename'] for file in mail.attachments]))

mail.write\_attachments('pop3\_attachments')

print('Отключение от сервера. Удаление сообщений.')

server.quit()

print('Выход...')

IMAP

import imaplib

import mailparser

import getpass

import sys

if \_\_name\_\_ == '\_\_main\_\_':

connect = imaplib.IMAP4\_SSL('imap.gmail.com')

login = input('email: ')

password = getpass.getpass('password: ')

if not login and not password:

login = 'silurcpppy@gmail.com'

password = 'smtplib1'

try:

print(f"connect: {connect.login(login, password)[1][0].decode()}")

except BaseException:

print('Wrong login or password.')

sys.exit(-1)

result, boxes = connect.list()

print(f'a result of list: {result}')

for box in boxes:

print(box.decode('utf-8'))

print(f"Select INBOX: {connect.select('INBOX')}")

result, data = connect.uid('search', None, 'ALL')

print(f"a result of search: {result} uids: ", data)

uids = data[0].split()[-1:-11:-1]

print(f'uids = {uids}\n')

for i in uids:

result, data = connect.uid('fetch', i, '(RFC822)')

print(f'Message: {str(i)}')

parse = mailparser.parse\_from\_bytes(data[0][1])

print(f"From: {' '.join(parse.from\_[0])}")

print(f"To: {parse.delivered\_to}")

print(f"Subject: {parse.subject}")

print(f'Дата: {parse.date.replace(hour=parse.date.hour + 3)}')

print('Text:\n' + ''.join(parse.text\_plain))

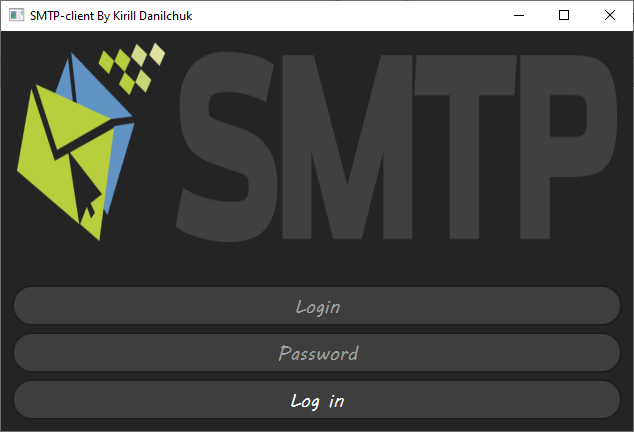
print('Files:\n', ' '.join([file['filename'] for file in parse.attachments]))

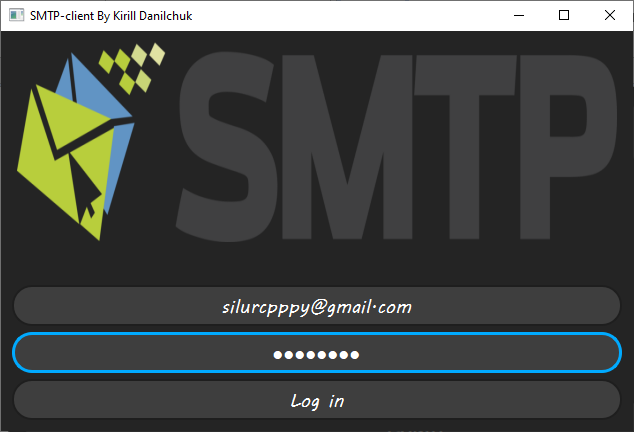
print('-' \* 30, '\n')

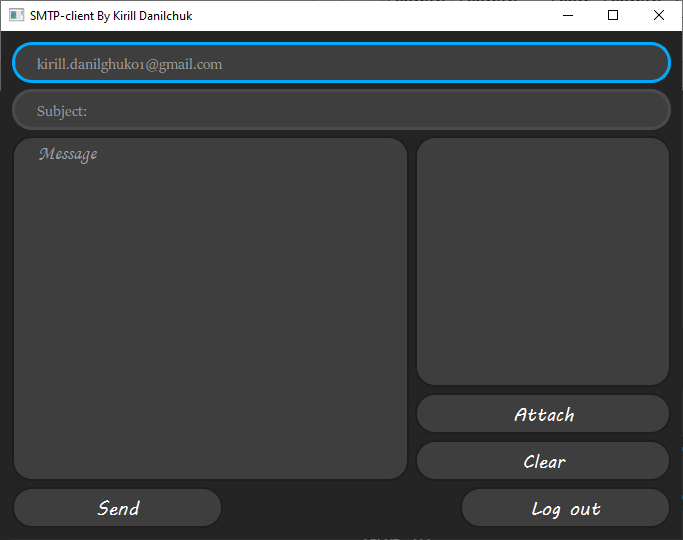
parse.write\_attachments('imap\_attachments')

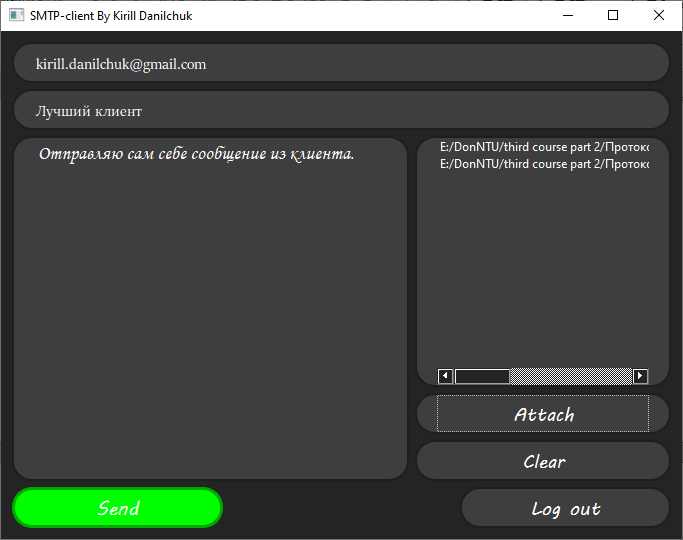
СКРИНШОТЫ

SMTP

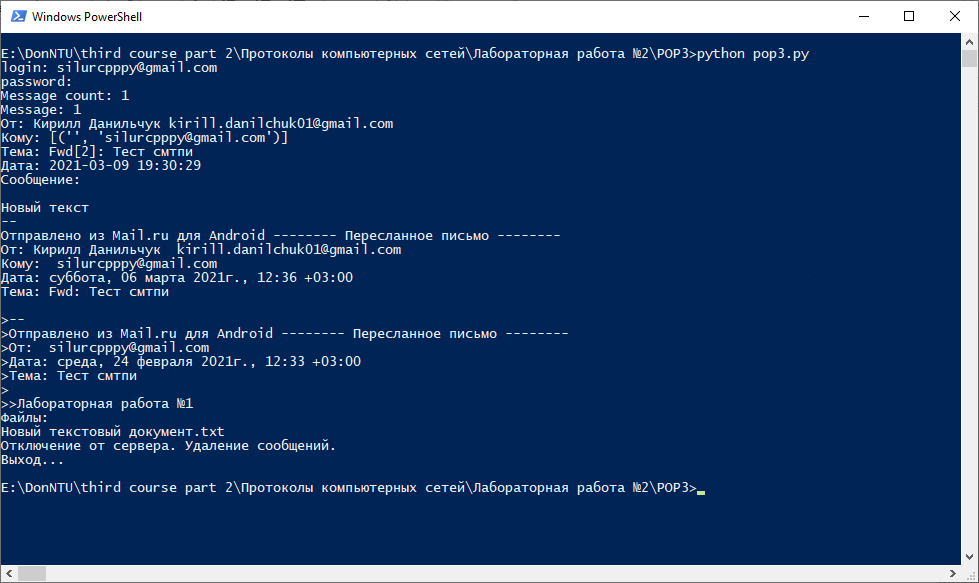


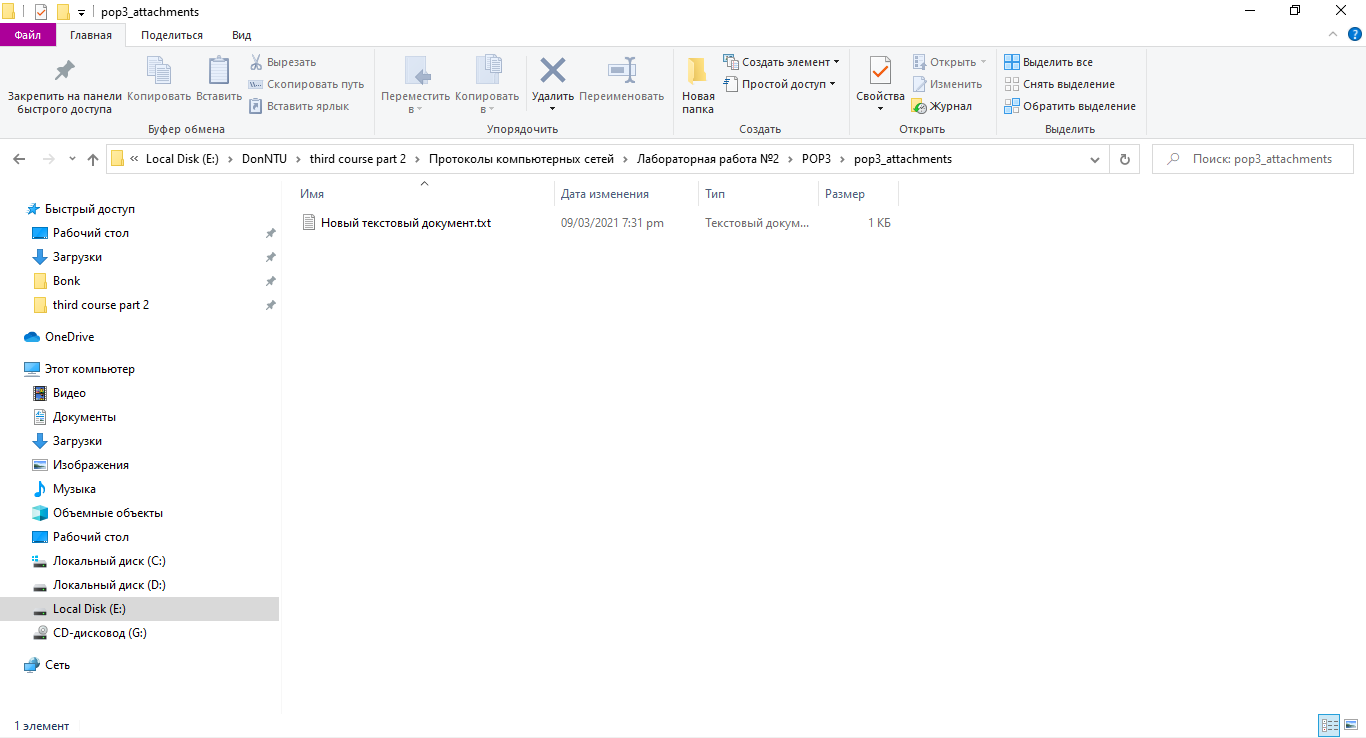


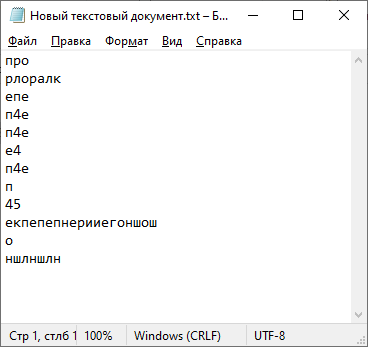




POP3







IMAP

