|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

*ФАКУЛЬТЕТ «Информатика и системы управления»*

*КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»*

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 05 |

**“Настройка сетевых служб: DNS, HTTP, электронной почты в сетевом эмуляторе”**

**Дисциплина:  *Компьютерные сети***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ***ИУ7И-76Б*** |  |  | **Нгуен Ф. С.** |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | |  | | --- | | **Рогозин Н. О.** | |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

*Москва, 2021*

**Задание**: Написать smtp-клиент, который

1) В качестве входных данных (аргументы командной строки) получает: адрес получателя, адрес отправителя, пароль.

2) Использует один из открытых smtp-серверов для доставки MIME-сообщений, включая приложения, если они есть, в соответствии с вариантом.

Вариант = номер студента по списку в Электронном Университете % кол-во вариантов.

3) Дополнительная задача зависит от варианта.

Доставка сообщений выполняется с регулярным интервалом. Интервал и тело сообщения, имя файла для прикрепления (опционально) вводятся с клавиатуры.

Допускается использование любого ЯП и фреймворков.

* **Получает: адрес получателя, адрес отправителя, пароль.**

ar = sys.argv

receiver\_email = ar[1]

sender\_email = ar[2]

password = ar[3]

$ python main.py [nguyensangqqhh@gmail.com](mailto:nguyensangqqhh@gmail.com) [nguyensanghso@gmail.com](mailto:nguyensanghso@gmail.com) mypassword

* **Создание составного сообщения**

message = MIMEMultipart()

message["From"] = sender\_email

message["To"] = receiver\_email

message["Subject"] = subject

message.attach(MIMEText(body, "plain"))

* **Файл для прикрепления**

try:

with open(filename, "rb") as attachment:

part = MIMEBase("application", "octet-stream")

part.set\_payload(attachment.read())

encoders.encode\_base64(part)

part.add\_header(

"Content-Disposition",

f"attachment; filename= {filename}",

)

message.attach(part)

except:

if c == None:

print("File [{}] not Found".format(filename))

c = input("Send Mail without file?[Y/N]")

if (c == 'Y' or c == 'y'):

None

else:

exit(-1)

* **Запуск безопасного SMTP- серверa (SSL port = 465)**

with smtplib.SMTP\_SSL("smtp.gmail.com", 465) as server:

server.login(sender\_email, password)

server.sendmail(sender\_email, receiver\_email, text)

server.quit()

* **Доставка сообщений выполняется с регулярным интервалом.**

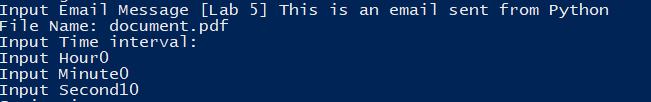
for i in range(n):

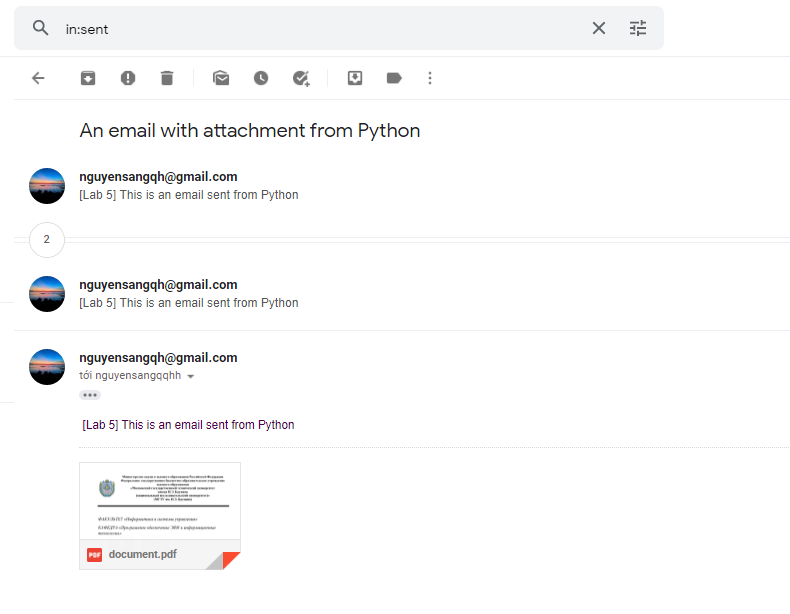
send\_mail()

time.sleep(time\_interval)

**Результат**

$ python main.py nguyensangqqhh@gmail.com nguyensangqh@gmail.com mypassword





**Программный код**

**import** sys

**import** email**,** smtplib**,** ssl

**import** time

**from** email **import** encoders

**from** email**.**mime**.**base **import** MIMEBase

**from** email**.**mime**.**multipart **import** MIMEMultipart

**from** email**.**mime**.**text **import** MIMEText

**def** send\_mail**():**

**global** subject**,** body**,** sender\_email**,** receiver\_email**,** password**,** filename**,** c

# Create a multipart message and set headers

message **=** MIMEMultipart**()**

message**[**"From"**]** **=** sender\_email

message**[**"To"**]** **=** receiver\_email

message**[**"Subject"**]** **=** subject

# Add body to email

message**.**attach**(**MIMEText**(**body**,** "plain"**))**

**try:**

# Open PDF file in binary mode

**with** open**(**filename**,** "rb"**)** **as** attachment**:**

# Add file as application/octet-stream

# Email client can usually download this automatically as attachment

part **=** MIMEBase**(**"application"**,** "octet-stream"**)**

part**.**set\_payload**(**attachment**.**read**())**

# Encode file in ASCII characters to send by email

encoders**.**encode\_base64**(**part**)**

# Add header as key/value pair to attachment part

part**.**add\_header**(**

"Content-Disposition"**,**

f"attachment; filename= {filename}"**,**

**)**

# Add attachment to message and convert message to string

message**.**attach**(**part**)**

**except:**

**if** c **==** **None:**

**print(**"File [{}] not Found"**.**format**(**filename**))**

c **=** input**(**"Send Mail without file?[Y/N]"**)**

**if** **(**c **==** 'Y' **or** c **==** 'y'**):**

**None**

**else:**

exit**(-**1**)**

text **=** message**.**as\_string**()**

# Log in to server using secure context and send email

#context = ssl.create\_default\_context()

**with** smtplib**.**SMTP\_SSL**(**"smtp.gmail.com"**,** 465, context = context**)** **as** server**:**

server**.**login**(**sender\_email**,** password**)**

server**.**sendmail**(**sender\_email**,** receiver\_email**,** text**)**

server**.**quit**()**

**#main**

**if** \_\_name\_\_ **==** "\_\_main\_\_"**:**

**try:**

ar **=** sys**.**argv

receiver\_email **=** ar**[**1**]**

sender\_email **=** ar**[**2**]**

password **=** ar**[**3**]**

subject **=** "An email with attachment from Python"

body **=** input**(**"Input Email Message”)

filename **=** input**(**"File Name: "**)**

**print(**"Input Time interval:"**)**

h **=** int**(**input**(**"Input Hour"**))**

m **=** int**(**input**(**"Input Minute"**))**

s **=** int**(**input**(**"Input Second"**))**

time\_interval **=** s **+** m **\*** 60 **+** h **\*** 3600

**except:**

exit**(-**1**)**

c **=** **None**

n **=** 5

**print(**"Sending..."**)**

**for** i **in** range**(**n**):**

send\_mail**()**

time**.**sleep**(**time\_interval**)**

**print(**"Sent"**)**